






# FINAL: Combining First-Order Logic With Natural Logic for Question Answering

Jihao Shi , *Student Member, IEEE*, Xiao Ding , Siu Cheung Hui , Yuxiong Yan, Hengwei Zhao , Ting Liu, and Bing Qin 

**Abstract**—Many question-answering problems can be approached as textual entailment tasks, where the hypotheses are formed by the question and candidate answers, and the premises are derived from an external knowledge base. However, current neural methods often lack transparency in their decision-making processes. Moreover, first-order logic methods, while systematic, struggle to integrate unstructured external knowledge. To address these limitations, we propose a neuro-symbolic reasoning framework called Final, which combines First-order logic with Natural Logic for question answering. Our framework utilizes *first-order logic* to systematically decompose hypotheses and *natural logic* to construct reasoning paths from premises to hypotheses, employing bidirectional reasoning to establish links along the reasoning path. This approach not only enhances interpretability but also effectively integrates unstructured knowledge. Our experiments on three benchmark datasets, namely QASC, WorldTree, and WikiHop, demonstrate that FINAL outperforms existing methods in commonsense reasoning and reading comprehension tasks, achieving state-of-the-art results. Additionally, our framework also provides transparent reasoning paths that elucidate the rationale behind the correct decisions.

**Index Terms**—Natural logic, first-order logic, textual inference, question answering.

## I. INTRODUCTION

QUESTION Answering (QA) is a crucial task in natural language processing, involving the retrieval and synthesis of information to answer questions posed in natural language. Prior work [1] has approached question answering as a form of textual entailment, where the external corpus is treated as a collection of premises, and the question and candidate answers are combined to form hypotheses. A significant challenge in this approach is that the external knowledge corpus does not explicitly answer all questions, necessitating the inference of many implicit answers. Furthermore, answering a question often

requires composing multiple pieces of knowledge to form a conclusive hypothesis [2], [3], [4], [5].

Neural networks, particularly pre-trained language models (PLMs) such as BERT, RoBERTa, and DeBERTa [6], [7], [8], have shown strong performance in various QA scenarios. However, a significant downside of these models is their lack of interpretability. As “black-box” models, they provide decision results without any explanation of how those results were reached, making it challenging for humans to understand and trust their decisions. To address this issue, combining symbolic methods with neural networks presents a promising solution. By integrating first-order logic (FOL) into the reasoning process, we can provide a step-by-step explanation for the decisions made by neural networks. This approach harnesses the decomposition capabilities of first-order logic while preserving the robustness and high performance inherent in neural networks [9], [10], [11].

While first-order logic is valuable for decomposing hypotheses, it faces difficulties in converting natural language text from an external corpus into first-order logic. For example, the sentence “loud noises can cause mammals to startle” might be reduced to the tuple [loud noises, can cause, mammals] in FOL, failing to capture crucial semantic details. On the other hand, *Natural logic* [12], [13] uses deduction based on the monotonicity in calculus [14], [15], enabling reasoning directly on the surface form of natural language text without needing conversion to FOL. To overcome these challenges, we introduce **FINAL**, a neuro-symbolic framework that combines the strengths of **F**irst-order logic and **N**atural Logic to enhance differentiable reasoning. This combination allows us to systematically decompose hypotheses and construct reasoning paths from premises to hypotheses while preserving the semantic integrity of the text. The advantages of this combination are illustrated in Table I.

The proposed framework, **FINAL**, employs a bidirectional reasoning process. Initially, it transforms the question and candidate answers into hypotheses. Since the gold answer to the questions draws upon information from multiple texts, the framework utilizes first-order logic to break down the hypothesis into sub-hypotheses. Simultaneously, the framework employs natural logic to rewrite premises from the external corpus, resulting in intermediate premises that convey the same meaning as the original premise. This iterative process, performed by the natural logic module, generates multiple reasoning paths. To ultimately select the answer, **FINAL** calculates the semantic similarity between the intermediate premises and the sub-hypotheses,

Received 29 February 2024; revised 25 July 2024; accepted 9 March 2025. Date of publication 14 March 2025; date of current version 1 May 2025. This work was supported in part by the National Natural Science Foundation of China under Grant U22B2059 and Grant 62176079 and in part by the Natural Science Foundation of Heilongjiang Province under Grant YQ2022F005. Recommended for acceptance by Z. Wang. (Corresponding author: Xiao Ding.)

Jihao Shi, Xiao Ding, Yuxiong Yan, Hengwei Zhao, Ting Liu, and Bing Qin are with the Research Center for Social Computing and Information Retrieval, Harbin Institute of Technology, Harbin 150001, China (e-mail: jhshi@ir.hit.edu.cn; xding@ir.hit.edu.cn; yxian@ir.hit.edu.cn; hwzhao@ir.hit.edu.cn; tliu@ir.hit.edu.cn; qinb@ir.hit.edu.cn).

Siu Cheung Hui is with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798 (e-mail: ASSCHUI@ntu.edu.sg).

Digital Object Identifier 10.1109/TKDE.2025.3551231

TABLE I  
COMPARISON OF OUR PROPOSED METHOD TO OTHER APPROACHES BASED ON  
THREE IMPORTANT FEATURES: EXPLAINABILITY, DECOMPOSITION, AND  
SEMANTICS INTEGRITY

FEATURE	Methods			
	NN	FOL	NL	Ours
Explainability	✗	✓	✓	✓
Decomposition	✓	✓	✗	✓
Semantics Integrity	✓	✗	✓	✓

The abbreviation “NN” refers to neural networks, “FOL” represents first-order logic, and “NL” stands for natural logic.

using the similarity score as the determining factor. The entire framework is trained end-to-end, using the correct choice as the target. Moreover, the framework does not require any additional intermediate labels.

Our contributions can be summarized as follows:

- We propose **FINAL**, a novel framework that seamlessly integrates neural networks with first-order logic and natural logic for question answering. This hybrid approach leverages the strengths of both symbolic reasoning and neural models to enhance interpretability and performance.
- By combining first-order logic and natural logic, our proposed framework facilitates a systematic and explainable reasoning process. This integration breaks down complex hypotheses into manageable sub-hypotheses while preserving the semantic integrity of external knowledge. In addition, we also introduce a phrase-level mutation technique during natural logic reasoning, which shortens reasoning paths and improves verb phrase alignment.
- We conduct comprehensive evaluations on three benchmark datasets, including QASC, WorldTree, and WikiHop. Our results demonstrate that **FINAL** outperforms existing methods in commonsense reasoning and reading comprehension tasks, achieving state-of-the-art performance. Moreover, our framework provides transparent reasoning paths that elucidate the decision-making process.

The remainder of the paper is structured as follows. Section II defines the problem and provides background knowledge. Section III details our proposed framework, FINAL. Section IV describes the experimental setup, including datasets, metrics, baselines, and implementation details. Section V presents the results and highlights the advantages of FINAL over existing methods. Section VI reviews related works. Finally, Section VII concludes the paper and discusses future directions.

## II. PRELIMINARIES

In this section, we start by presenting the task and then introduce the foundational concepts of first-order logic and natural logic.

### A. The Problem

Let’s take an example of a multiple-choice question sourced from [5]. The question is presented below for reference, with the correct answer being underlined:

*Example-1:*

*Question:* What can cause a forest fire?

(A) trailers and boats (B) static electricity(C) plasma and formed elements (D) being over land

(E) thermal expansion (F) a surface not sterilized

(G) microbes and mussels (H) two or more sets of alleles

*Knowledge Base:*

...Static electricity can be the cause of sparks. Sparks from cigarette butts frequently ignite forest fires ...

In this study, the objective is to develop a model that can accurately identify the correct answer to a multiple-choice question based on an unstructured knowledge base. Following the approach proposed in [16], we treat the multiple-choice question-answering task as a textual entailment problem. To begin, we convert both the question and the eight candidate answers into declarative sentences, referred to as target hypotheses  $S_i$ , where  $i$  ranges from 1 to 8. Our next step involves retrieving a relevant set of fact pairs, denoted as  $\mathcal{F} = \{(f_1, f_2) \mid f_1 \in \mathcal{K}, f_2 \in \mathcal{K}\}$ , from the unstructured knowledge base  $\mathcal{K}$ . The aim is to identify a specific pair from  $\mathcal{F}$  that supports one of the eight hypotheses. To achieve this, our approach focuses on the development of a neural-symbolic model, which utilizes first-order logic to decompose the hypothesis and employs natural logic as the underlying prover to facilitate the construction of the proving process.

### B. First-Order Logic

First-order logic is a widely used symbolic method in the neuro-symbolic framework. Within this framework, we are particularly interested in a subset of statements called *Horn clauses* [17], [18], [19]. These logical formulas have a specific chain-like structure and have been utilized in several studies [9], [20], [21], [22]. A Horn clause has a general format as follows:

$$p(X, Z) \leftarrow p_1(X, Y_1) \wedge \cdots \wedge p_k(Y_k, Z), \quad (1)$$

where  $p$  and  $p_k$  are predicate symbols, while  $X$ ,  $Y_k$ , and  $Z$  are variable symbols that can be instantiated by words or phrases. The clause consists of a *head*, represented by  $p(X, Z)$ , and a *body*, denoted by  $p_1(X, Y_1) \wedge \cdots \wedge p_k(Y_k, Z)$ . The size of the body is indicated by  $k$ .

In our context, an *atom* refers to a predicate and its arguments. For instance, we consider  $p(X, Z)$  as an atom that can be instantiated as *can\_cause(static electricity, a forest fire)*. Here, the variables  $X$  and  $Z$  are assigned with specific values “static electricity” and “a forest fire”, respectively. The predicate  $p$  is associated with the concept of “can\_cause”.

### C. Natural Logic

Natural logic [12] is a formal proof theory that allows for direct reasoning on the surface form of natural language. It is based on the principles of monotonicity or projectivity [13], [23]. In natural logic proving, the main operations involve word-level insertion, deletion, and mutation, which are performed based on contextual monotonicity. The core reasoning rule of natural logic relies on contextual monotonicity and the logical relation of word-level mutation. MacCartney and Manning [13] proposed

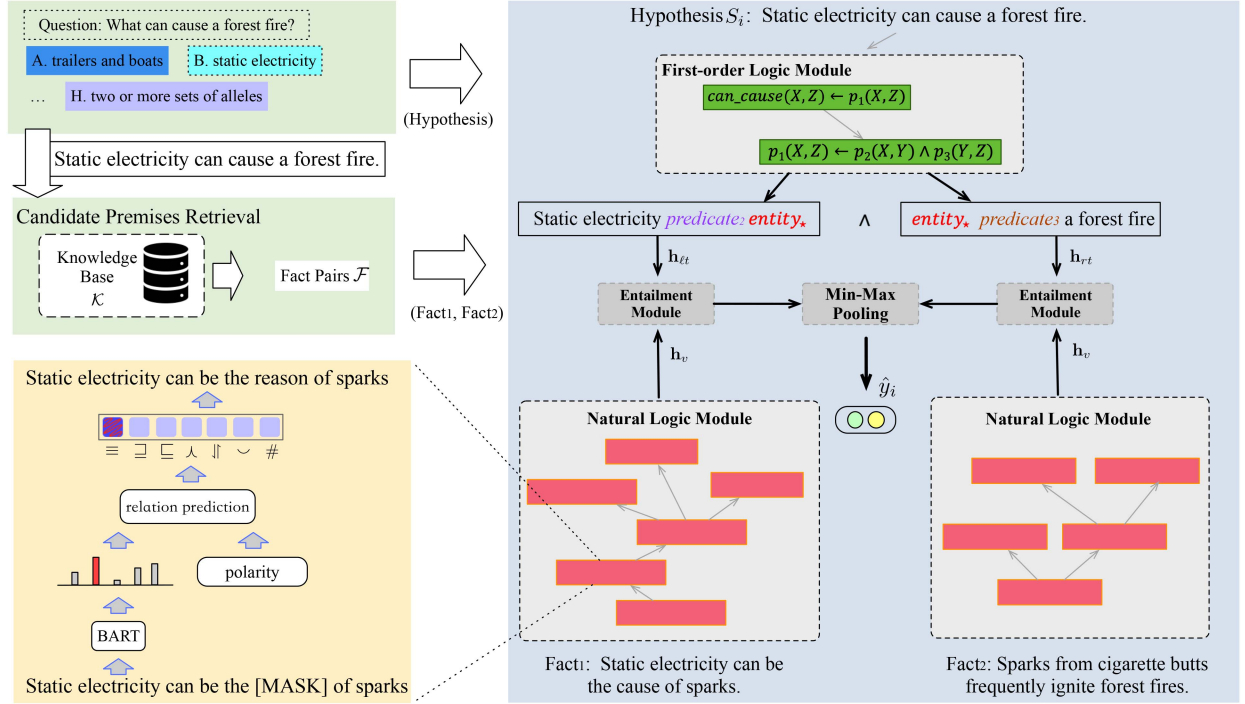


Fig. 1. Overview of the proposed FINAL framework for question answering.

TABLE II  
A SET  $\mathcal{D}$  OF SEVEN LOGICAL RELATIONS PROPOSED  
BY MACCARTNEY AND MANNING [13]

Relation	Name	Example
$x \equiv y$	equivalence	shrub $\equiv$ bush
$x \sqsubseteq y$	forward entailment	shrub $\sqsubseteq$ plant
$x \supseteq y$	reverse entailment	plant $\supseteq$ shrub
$x \wedge y$	negation	usual $\wedge$ unusual
$x \uparrow y$	alternation	shrub $\uparrow$ oak
$x \sim y$	cover	male $\sim$ doctor
$x \# y$	independence	shrub $\#$ luck

a set of seven relations, namely  $\mathcal{D} = \{\equiv, \sqsubseteq, \supseteq, \wedge, \uparrow, \sim, \#\}$ , as shown in Table II. For instance, mutating the word “plants” to “shrubs” corresponds to a reverse entailment relation, i.e., “plants”  $\supseteq$  “shrubs”. Natural logic then projects the logical relation based on the monotonicity or projectivity determined by the context. According to the monotonicity in calculus, an upward monotone preserves the logical relation, while a downward monotone can change the logical relation. For example, the quantifier “all” is a downward monotone in its first argument. Therefore, given “plants”  $\supseteq$  “shrubs”, we can infer that “all plants”  $\sqsubseteq$  “all shrubs” (e.g., as in the sentence “all plants need water to survive”  $\sqsubseteq$  “all shrubs need water to survive”).

### III. METHOD

In this paper, we propose a novel neuro-symbolic framework called FINAL, which combines first-order logic with natural logic to enable differentiable reasoning. The architecture of our proposed framework FINAL is illustrated in Fig. 1.

#### A. Knowledge Selection

Starting with a question sentence, such as “What can cause a forest fire” and a candidate answer like “static electricity”, FINAL transforms them into a declarative hypothesis sentence  $S_i$ , which in this case would be “static electricity can cause a forest fire”.

Answering a compositional question necessitates gathering information from multiple pieces of external knowledge. This involves retrieving relevant fact pairs from the external knowledge base  $\mathcal{K}$ . In this scenario, the hypothesis  $S_i$  serves as the starting point for FINAL to initiate the retrieval process. To accomplish this, a two-step information retrieval (IR) process, similar to the approach presented in [5] that leverages ElasticSearch [24], is employed. The first step involves retrieving fact  $f_1$  from the knowledge base. Subsequently, fact  $f_2$  is retrieved based on fact  $f_1$ . We impose a condition that  $f_2$  must contain at least one word from  $S_i$  but not from  $f_1$  (formally,  $\{word | word \in S_i - f_1\}$ ), as well as one word from  $f_1$  not from  $S_i$  (formally,  $\{word | word \in f_1 - S_i\}$ ). This criterion ensures that  $f_2$  is contextually relevant to both  $S_i$  and  $f_1$ .

To elucidate this process, let’s consider an example where  $S_i$  (the hypothesis) is “Static electricity can cause a forest fire” and the fact  $f_1$  (the first retrieved fact) is “Static electricity can be the cause of sparks”. The set difference of  $S_i$  and  $f_1$  includes words unique to  $S_i$ , such as “forest fire”, and the set difference of  $f_1$  and  $S_i$  consists of words unique to  $f_1$ , such as “sparks”. We then search for a fact  $f_2$  that contains at least one of these distinctive words to establish a connection between the hypothesis and the initially retrieved fact knowledge. Assume ElasticSearch retrieves the fact  $f_2$  “Sparks from cigarette butts frequently ignite forest fires”.



frequently ignite forest fires”. This fact  $f_2$  meets our criterion because it contains “forest fire” from the set difference of  $S_i$  and  $f_1$ , and “sparks” from the set difference of  $f_1$  and  $S_i$ . By ensuring that  $f_2$  contains words from both set differences, we maintain the contextual relevance between  $S_i$ ,  $f_1$ , and  $f_2$ . Notably, the retrieved fact  $f_2$  must contain at least one word from the set difference of  $S_i$  and  $f_1$ , as well as the set difference of  $f_1$  and  $S_i$ . Finally, the top  $\mu$  fact pairs are selected based on the sum of individual IR scores. The value of  $\mu$  is tuned using the development set.

### B. Neural First-Order Reasoning

**Triple Extraction:** To begin, we start by converting the hypothesis  $S_i$  into a triple format. This can be achieved by identifying the noun phrases present in the hypothesis. In our case, we utilize the noun chunks extraction technique provided by [25]. For instance, let’s consider the hypothesis “*Static electricity can cause a forest fire*”. This sentence contains two noun phrases, namely “*static electricity*” and “*a forest fire*”. The remainder of the sentence can be regarded as the predicate. Therefore, we can represent the hypothesis in the triple format as [*static electricity*, *can cause*, *a forest fire*].

**First-order Logic Module:** We decompose the hypothesis into a triple format using the following rule:

$$p_1(X, Z) \leftarrow p_2(X, Y) \wedge p_3(Y, Z), \quad (2)$$

where  $X$ ,  $Y$ , and  $Z$  are variables that can be instantiated by words or phrases. The notation  $p_n$  stands for  $predicate_n$ , which represents the predicate. In this paper, we focus on the rule which consists of two atoms in the body. Our method can be easily applied to rules with any number of atoms in the body. When we conduct decomposition, we generate two new tuples  $p_2(X, Y)$  and  $p_3(Y, Z)$ . Intuitively, these new tuples inherit the arguments  $X$  and  $Z$  from the rule head, and they are connected by a shared argument  $Y$ . As a result, we only need to predict three tokens - two predicates and one shared argument. For instance, let’s consider the example [*static electricity*, *can cause*, *a forest fire*]. This can be decomposed into two new sub-hypotheses: [*static electricity*,  $predicate_2$ ,  $entity_*$ ] and [ $entity_*$ ,  $predicate_3$ , *a forest fire*]. Taking inspiration from Neural Theorem Provers [9], we introduce a neural network-based reasoning module to conduct chain-like logical reasoning.

In order to obtain the representation for each token, we utilize the pre-trained language model DeBERTa [8] as the encoder. This model effectively captures the relative positions of words in a sentence, which results in more accurate token representations and more effective modeling of the input sequence for contextual semantic representations. For all input tokens  $S_i = ([CLS], w_1, \dots, w_L)$ , we compute the hidden states  $\mathbf{H}_{enc} = (\mathbf{h}_{[CLS]}, \mathbf{h}_1, \dots, \mathbf{h}_L)$  using the following equation:

$$\mathbf{H}_{enc} = \text{DeBERTa}(\phi^{emb}(S_i)), \quad (3)$$

where  $L$  represents the number of tokens in the hypothesis, and  $\phi^{emb}$  refers to the embedding layer of DeBERTa.

The vector representations of the two generated predicates are computed as follows:

$$\begin{aligned} \mathbf{h}'_{p_1} &= \text{avg}(\mathbf{h}_{p_1}, \mathbf{h}_{[CLS]}), \\ \mathbf{h}_{p_2} &= \text{SiLU}(\mathbf{W}_{p_2} \mathbf{h}'_{p_1} + b_{p_2}), \\ \mathbf{h}_{p_3} &= \text{SiLU}(\mathbf{W}_{p_3} \mathbf{h}'_{p_1} + b_{p_3}), \end{aligned} \quad (4)$$

where  $\mathbf{h}_{p_1}$ ,  $\mathbf{h}_{p_2}$  and  $\mathbf{h}_{p_3}$  represent the vector representations of  $predicate_1$ ,  $predicate_2$  and  $predicate_3$ , respectively.  $\mathbf{h}_{[CLS]}$  refers to the vector representation of the entire sentence.  $\text{avg}(\cdot, \cdot)$  denotes the mean average function. The parameters  $\mathbf{W}_{p_2}$ ,  $\mathbf{W}_{p_3}$ ,  $b_{p_2}$  and  $b_{p_3}$  are learnable, and SiLU [26] is the swish function. The vector representation of the shared argument  $\mathbf{h}_{entity_*}$  is computed as follows:

$$\begin{aligned} \mathbf{h}_{avg} &= \text{avg}(\mathbf{h}_{[CLS]}, \mathbf{h}_X, \mathbf{h}_{p_1}, \mathbf{h}_Z), \\ \mathbf{h}_{entity_*} &= \text{SiLU}(\mathbf{W}_e \mathbf{h}_{avg} + b_e), \end{aligned} \quad (5)$$

where  $\mathbf{h}_X$ ,  $\mathbf{h}_{p_1}$  and  $\mathbf{h}_Z$  represent the vector representations of the argument  $X$ , predicate  $predicate_1$  and argument  $Z$ , respectively. The function  $\text{avg}(\cdot, \cdot, \cdot, \cdot)$  calculates the mean average of the input vectors. The parameters  $\mathbf{W}_e$  and  $b_e$  are learnable parameters. Note that the vector  $\mathbf{h}_{entity_*}$  represents an entity in the embedding space. To decode the entity, we can calculate the nearest Euclidean distance to  $\mathbf{h}_{entity_*}$  in the vector space of the pre-selected entity base. Upon obtaining the vector representation of the generated predicates and the shared argument, we can calculate the vector representation of the two generated tuples. For convenience, we refer to these two tuples as the left sub-hypothesis  $hyp_l$  and the right sub-hypothesis  $hyp_r$ . The representations of these two sub-hypotheses are computed as follows:

$$\begin{aligned} \mathbf{h}_{\ell t} &= \text{mean}(\mathbf{h}_X, \mathbf{h}_{p_2}, \mathbf{h}_{entity_*}), \\ \mathbf{h}_{rt} &= \text{mean}(\mathbf{h}_{entity_*}, \mathbf{h}_{p_3}, \mathbf{h}_Z), \end{aligned} \quad (6)$$

where  $\mathbf{h}_{\ell t}$  and  $\mathbf{h}_{rt}$  represent the vector representations of the two generated tuples  $p_2(X, Y)$  and  $p_3(Y, Z)$ , respectively.

### C. Neural Natural Logic Reasoning

**Candidate Path Generation:** We utilize a forward proof process to generate paths that closely align with the sub-hypotheses generated by first-order logic reasoning. This involves following natural logic inference steps to ensure that the original premise entails the intermediate premises. Once we establish semantic closeness, we can then prove that the hypothesis can be entailed by the fact pair through the inference steps.

For convenience, we will describe how to conduct neural natural logic reasoning from  $fact_1$  to the left sub-hypothesis tuple. The same reasoning method can be applied from  $fact_2$  to the right sub-hypothesis tuple. When performing mutations, it is not necessary to mutate each word in the premise. Therefore, we use spaCy [25] to obtain part-of-speech (POS) tags and instruct FINAL to ignore tokens with the following POS tags: prepositions, determiners, coordinating conjunctions, cardinal numbers, personal pronouns, and modal verbs. Additionally, since some predicates are in the form of phrases and phrase-level

reasoning can shorten the reasoning path, we utilize BART [27] as the underlying model to decode one or more tokens for each mask.

We conduct inference starting from the fact  $f_1$ , which can be considered as the premise. This fact consists of  $L$  words, denoted as  $f_1 = (w_1, \dots, w_\ell, \dots, w_L)$ . To conduct mask filling, a word in the premise is masked and then fed into BART, as shown in the left bottom part of Fig. 1. In order to narrow down the semantic closeness with the left sub-hypothesis, we also perform inference through insertion and deletion operations. Following Shi et al. [28], we only insert or delete adjectives or adjective phrases. The insertion operation uses the mask mechanism by inserting a [MASK] token in front of a noun. We keep track of the position of insertion/mutation/deletion to avoid repeating operations in the same position. The list of candidate words or phrases is obtained through the output of BART, which takes into consideration the context instead of directly searching from WordNet [29].

*Pruning Candidate Paths:* To ensure that the intermediate premise can be deduced from the original premise, it is important to follow the rules of natural logic reasoning. This involves using the monotonicity tags and the logical relation prediction module to determine whether the mutation operation changes the meaning of the original premise. By doing so, we can filter out any incorrect mutations.

In order to determine the polarity of each word, we utilize the *natlog* parser,<sup>1</sup> which tags each word as either upward or downward monotone. After determining the polarity, we predict the logical relation between the original token and the generated token using a fine-tuned RoBERTa model, as described in [28]. By inputting the original token  $w_\ell$  and the generated token  $c_\ell$  into the RoBERTa model, we can compute the probabilistic distribution  $P_r$  over the seven natural logical relations listed in Table II. The formula for the calculation is as follows:

$$P_r = \text{softmax}(f(w_\ell, c_\ell)), \quad (7)$$

where  $f(\cdot)$  represents the RoBERTa model. The input format for this model is [CLS]  $w_\ell$  [SEP]  $c_\ell$  [SEP]. Here,  $w_\ell$  refers to the original token, while  $c_\ell$  represents the generated token. The logical relation with the highest score,  $\max(P_r)$ , is considered as the predicted logical relation.

Then, we use a projection function  $\xi$  to derive the sentence-level logical relation based on the monotonicity of the word  $w_\ell$  and the predicted logical relation  $r$ . When the monotonicity is upward monotone, the sentence-level logical relation remains the same as the predicted logical relation. However, if the monotonicity is downward monotone, we can predict the sentence-level logical relation using the projection operator  $\xi$  as shown in Table III. To ensure that the original premise's semantics remain unchanged, we only perform inference on the sentence-level relations of *equivalence* ( $\equiv$ ) and *forward entailment* ( $\sqsubseteq$ ).

For *reverse entailment* ( $\supseteq$ ), *cover* ( $\sim$ ) and *independence* ( $\#$ ) relations, these correspond to unknown validity of inference. In some situations, they may perform valid inference, while in other situations, they may perform invalid inference. To ensure

TABLE III  
PROJECTION FUNCTION  $\xi$  FOR DOWNWARD MONOTONICITY  
POLARITY OF THE MUTATED WORD

$r$	$\equiv$	$\sqsubseteq$	$\supseteq$	$\wedge$	$\Downarrow$	$\sim$	$\#$
$\xi(r)$	$\equiv$	$\supseteq$	$\sqsubseteq$	$\Downarrow$	$\wedge$	$\sim$	$\#$

Input  $r$  represents the predicted logical relation.

the original semantics are unchanged, we discard any mutations involving unknown validity of inference that could potentially introduce logical inconsistencies or alter the original semantics. The logical relations *negation* ( $\wedge$ ) and *alternation* ( $\Downarrow$ ) represent mutual exclusivity. When handling these two logical relations, any mutations should respect this exclusivity. For instance, if the predicted logical relation between the original token and the generated token is *negation*, mutating the original with the generated token would violate the semantics of the original premise. Thus, our approach avoids any mutations that would introduce *negation* or *alternation* relationships in a manner that conflicts with the original semantics.

One approach to training the natural logic module is through end-to-end training. However, this method can lead to a large search space for inference, which can be inefficient. As a solution, we can pre-process the natural logic module, which helps reduce the complexity of the reasoning process in our FINAL framework. Specifically, we extend the proof process by performing phrase-level insertion, mutation, and deletion in a depth-first manner until reaching the maximum depth  $\lambda$ .

#### D. Merge

To evaluate the semantic closeness between the sub-hypotheses decomposed by first-order logic and the intermediate premises inferred through natural logic, we employ the DeBERTa model [8] to learn this semantic closeness. DeBERTa has demonstrated superior performance in understanding semantic nuances in text, making it particularly well-suited for semantic similarity calculation. The success of the proof is determined by the extent to which the sentences derived from both first-order logic and natural logic are semantically close in either branch. The computation of semantic closeness is performed as follows:

$$\begin{aligned} score_{\ell t} &= \max_{\forall v \in \mathcal{V}_{\ell t}} \text{sim}(\mathbf{h}_{\ell t}, \mathbf{h}_v), \\ score_{rt} &= \max_{\forall v \in \mathcal{V}_{rt}} \text{sim}(\mathbf{h}_{rt}, \mathbf{h}_v), \\ \hat{y}_i^+ &= \sigma(\min(score_{\ell t}, score_{rt})), \\ \hat{y}_i^- &= 1 - \hat{y}_i^+, \end{aligned} \quad (8)$$

where  $\mathbf{h}_{\ell t}$  and  $\mathbf{h}_{rt}$  represent the representations of the sub-hypotheses using first-order logic.  $\mathbf{h}_v$  represents the representation of intermediate premises using natural logic.  $\mathcal{V}_{\ell t}$  is the set of premises inferred by premise  $f_1$ , and  $\mathcal{V}_{rt}$  is the set of premises inferred by premise  $f_2$ . The function  $\text{sim}(\cdot)$  is used to calculate the semantic closeness. The sigmoid function  $\sigma$  is used to convert the scores into probabilities.  $\hat{y}_i^+$  refers to the probability of the positive class, and  $\hat{y}_i^-$  refers to the probability of negative class. In this context, the semantic closeness can

<sup>1</sup> <https://stanfordnlp.github.io/CoreNLP/natlog.html>

TABLE IV  
STATISTICS OF THE QASC, WORLDTREE AND WIKIHOP DATASETS

Train-Dev-Test Split	QASC	WorldTree			WikiHop			
		Easy	Challenge	Overall	Publisher	Developer	Country	Record_label
Train (# of questions)	8,134	708	278	986	509	267	742	2,305
Dev (# of questions)	926	158	68	226	54	29	194	283
Test (# of questions)	920	859	388	1,247	-	-	-	-

be interpreted as the supporting probability of the fact pair  $(f_1, f_2)$ , which corresponds to the probability of the corresponding answer. To determine the correct answer among all candidate choices, we apply the same procedure to all candidate choices and select the highest score  $\max_i(\hat{y}_i^+)$ . Ultimately, the answer that corresponds to the hypothesis with the highest scoring proof is selected.

#### E. Loss

For our proposed method, we construct  $N$  different input sequences using the question and each choice. We consider the input sequence corresponding to the gold answer  $a_{gold}$  as the positive class, while all other sequences are considered as negative classes. For each input sequence, we obtain the probabilities of the negative and positive classes, denoted as  $\hat{y}_i^-$  and  $\hat{y}_i^+$ , respectively. As a result, the loss function takes the following forms:

$$\begin{aligned}\mathcal{L} &= - \sum_{i=1}^N (t_i^+ \log(\hat{y}_i^+) + t_i^- \log(\hat{y}_i^-)), \\ &= - \log(\hat{y}_{gold}^+) - \sum_{i=1, i \neq gold}^N \log(\hat{y}_i^-),\end{aligned}\quad (9)$$

where  $t_i^+$  and  $t_i^-$  represent the class labels, and  $t_i^+ = 1, t_i^- = 0$  when the  $i$ -th candidate answer is correct; otherwise,  $t_i^+ = 0, t_i^- = 1$ .

### IV. EXPERIMENTAL SETUP

In this section, we discuss the experimental setup including the benchmark datasets, evaluation metrics, baseline models, and implementation details.

#### A. Benchmark Datasets and Evaluation Metrics

**Benchmark Datasets:** We perform experiments on different datasets that necessitate retrieving facts from an external corpus and synthesizing them to respond to multiple-choice questions. Table IV shows the statistics of the three datasets, namely QASC [5], WorldTree [30] and Wikihop [31], which are used in the experiments.

QASC [5] is a multiple-choice science question-answering dataset that tests commonsense and scientific inference. It consists of 9,980 questions with one correct answer and seven distractor answers, out of which four candidates are hard adversarial distractor answers. To answer these questions, we need to retrieve justification sentences from a knowledge base that contains 17.2 million facts [5]. WorldTree [30] also focuses

on multiple-choice science question answering. It consists of 2,459 questions with four choices, where only one choice is correct. The dataset is divided into a Challenge set and an Easy set. To answer the questions, we need to use a knowledge base that contains a large set of commonsense and scientific facts (around 10,000 facts) to construct explanations. WikiHop [31] is a multi-hop reading comprehension dataset. In this dataset, it is necessary to combine information from multiple documents to derive the correct answer. Each question consists of a query in the form of  $p(e, X)$ , where  $e$  is an entity,  $X$  is the entity that needs to be predicated from a list of candidate entities  $C$ , and  $p$  is the query predicate. We evaluate the dataset in four domains: *publisher*, *developer*, *country*, and *record\_label*. For each domain, we have a training set and a development set, with different numbers of questions. As the test set for WikiHop is not publicly available, we report the scores for the development set.

**Evaluation Metrics:** The model's performance is assessed using the accuracy scores as the evaluation metric across all datasets, as emphasized in previous studies [10], [28]. To assess the quality of the explanations, human evaluations are conducted. Three graduate students who are proficient in English language annotations are engaged to assign a score of either 0 or 1, evaluating the faithfulness of the provided inference path (explanation). Each explanation is independently scored by all three raters. The average explainability score, denoted as  $\overline{Score}_{expl}$ , is determined by the following formula:

$$\begin{aligned}\overline{Score}_{expl} &= \frac{\sum_{q=1}^M Annotation_q^{major}}{M}, \\ Annotation_q^{major} &= \arg \max_{a \in \{0,1\}} \sum_{u=1}^U \mathbb{1}\{a = Annotation_q^u\},\end{aligned}\quad (10)$$

where  $M$  represents the total number of random samples.  $Annotation_q^{major}$  denotes the majority annotation for the  $q$ -th sample.  $\mathbb{1}\{a = Annotation_q^u\}$  is the indicator function, evaluating to 1 when  $a = Annotation_q^u$ , and 0 otherwise.  $Annotation_q^u$  refers to the annotation provided by the  $u$ -th rater for the  $q$ -th sample, with values of 0 or 1.

To measure the quality of explainability scores annotated by human annotators, we calculate the inter-rater agreement using Fleiss' Kappa statistics [32]. The computation of the Kappa value  $\kappa$  is based on the following formula:

$$\kappa = \frac{P - P_e}{1 - P_e},$$



$$P = \sum_{q=1}^M \left( \frac{(\sum_j^J n_{qj}^2) - U}{MU(U-1)} \right),$$

$$P_e = \sum_j^J \left( \frac{\sum_{q=1}^M n_{qj}}{MU} \right)^2, \quad (11)$$

where  $P$  denotes the proportion of agreement among the raters.  $P_e$  represents the expected proportion of agreement under random annotation conditions.  $M$  is the total number of random samples,  $U$  stands for the total number of raters,  $J$  indicates the total number of manual annotation categories, which in this case is binary, denoted by the values 0 and 1, and  $n_{qj}$  represents the count of raters who assign the  $q$ -th sample to manual annotation category  $j$ .

### B. Baseline Models

**Large Language Models:** To conduct our baseline experiments, we utilized two categories of Large Language Models (LLMs): proprietary LLMs accessed via an API, and open source LLMs. In the first category, we employed GPT-3.5 [33] and GPT-4 [34] through the OpenAI API as references. Specifically, we used the versions `gpt-3.5-turbo-1106` and `gpt-4-1106-preview`. For the second category, we utilized open source models including LLAMA 2 [35], LLAMA 3 [36], Vicuna [37], ChatGLM [38], Baichuan [39], Mistral [40], Gemma [41], Qwen [42], and Flan-T5 [43]. To ensure fair evaluation, we specifically used versions of these models with sizes comparable to our model, such as 6B, 7B, 8B, 11B, and 13B parameters. Moreover, we focused on the chat versions of these models as they have been fine-tuned for chat-style use cases, which aligns well with our instruction-style prompts. Throughout the experiments, we followed the recommended hyperparameter settings for all LLMs, with a temperature of 0 for reproducibility.

In QA, LLMs often use *cloze prompting*, where the model is conditioned on a question without the associated answer options, and its chosen option is the one assigned the highest probability after normalization. However, *cloze prompting* may not fully leverage the capabilities of LLMs in MCQA [44]. Inspired by *multiple choice prompting* [44], we used the prompts shown in Table V. We conducted experiments using the same prompt format for the QASC and WorldTree datasets on the WikiHop dataset. However, the experimental results were not satisfactory. We hypothesize that the WikiHop dataset requires specific knowledge. Therefore, we designed a new prompt by adding the knowledge as a part of it. Additionally, to better understand the triplet format of the query, we separated the query and candidates for clarity.

**Finetuned Models:** Apart from large language models, we also conducted comparisons with several fine-tuned models that were trained on the target datasets. Specifically, we utilized BERT<sub>large</sub> [6], RoBERTa<sub>large</sub> [7], and DeBERTa<sub>large</sub> [8] models. For these models, the input format consisted of the concatenation of the context ( $context_i$ ) and question, followed by the answer ( $answer_i$ ), all enclosed in special tokens. During inference, the model normalized the output scores for

TABLE V  
PROMPTS USED BY LARGE LANGUAGE MODELS

#### Prompt Template for QASC and WorldTree

Choose the best answer to the following multiple-choice question delimited by triple backticks. And give the reason for your choice. Provide them in JSON format with the following keys: Answer, Reason.

```[question with multiple choices]```

#### Prompt Template for WikiHop

Assuming you know the following knowledge.

**Knowledge:**```[knowledge]```.

Choose the best answer to the following query from the candidates. And give the reason for your choice. Provide them in JSON format with the following keys: Answer, Reason.

**Query:**```[query]```

**Candidates:**```[candidates]```

each ( $question, answer_i$ ) pair and selected the answer with the highest score. Furthermore, we included a strong baseline model called TEAM [45], which treats the task of multi-choice question answering as a binary classification problem. The TEAM model categorizes ( $question, true\ answer$ ) as positive instances and ( $question, false\ answer$ ) as negative instances. TEAM, which is based on the “DeBERTa-large” model, has achieved significant improvements on several datasets.

In addition, we also compared FINAL with two neural-symbolic baselines: NeuNLI [28] and DILR [46]. The NeuNLI model incorporates natural logic reasoning within deep learning architectures to enhance question-answering performance, while the DILR model first extracts query-related information and then performs logical reasoning using first-order logic. It is worth mentioning that the DILR model is the top-performing model on the WikiHop dataset.

### C. Implementation Details

We utilized the “DeBERTa-large” pre-trained language model [8] with a vector dimension of 1024. In the natural logic-based reasoning process, a maximum search depth of 7 and a limit of 4 relevant fact pairs were set. The implementation of the model and training algorithm was based on the Transformers library [47]. The models were fine-tuned using the AdamW optimizer [48] with an initial learning rate of 1e-6. The training was conducted for 50 epochs on A100 GPUs with a total of 80 G memory. For the QASC and WorldTree datasets, the model that performed the best on the development set was selected and evaluated on the test set.

## V. RESULTS AND ANALYSIS

In this section, we evaluate the performance of FINAL and provide a comprehensive analysis of the results obtained. In addition, we explore the impact of hyperparameters  $\mu$  and  $\lambda$ , and present a case study on explainable paths.

TABLE VI  
PERFORMANCE RESULTS ON THE QASC, WORLDTREE AND WIKIHOP DATASETS

Model	QASC Accuracy (%)	WorldTree Accuracy (%)			WikiHop Accuracy (%)				
		Easy	Challenge	Overall	Publisher	Developer	Country	Record_label	Average
Open source LLMs 6B/7B/8B									
Baichuan2-Chat-7B [39]	45.43	72.18	54.12	66.56	38.89	34.48	62.89	22.97	39.81
LLAMA 2-CHAT-7B [35]	33.26	64.38	45.88	58.62	37.04	24.14	41.75	42.76	36.43
Vicuna-7B [37]	48.37	75.20	58.76	70.09	51.85	48.28	71.13	20.49	47.94
ChatGLM2-6B [38]	41.52	76.37	56.70	70.25	42.59	37.93	72.68	40.64	48.46
ChatGLM3-6B [38]	56.20	81.72	68.04	77.47	72.22	58.62	73.71	59.72	66.07
Mistral-7B-Instruct-v0.2 [40]	60.76	82.42	74.23	79.87	64.81	51.72	72.68	64.66	63.47
Mistral-7B-Instruct-v0.3 [40]	58.91	82.77	76.03	80.67	64.81	51.72	77.32	67.84	65.43
Gemma-7B-it [41]	64.13	85.33	72.16	81.23	61.11	51.72	69.59	41.70	56.03
Gemma-1.1-7B-it [41]	66.20	90.45	81.70	87.73	62.96	48.28	71.65	38.16	55.26
Qwen-7B-Chat [42]	66.85	87.19	67.53	81.07	51.85	44.83	68.04	42.05	51.69
Qwen1.5-7B-Chat [42]	72.61	91.85	78.35	87.65	66.67	51.72	75.77	65.72	64.97
LLAMA 3-8B-INSTRUCT [36]	71.30	91.50	80.41	88.05	62.96	48.28	76.29	55.83	60.84
Open source LLMs 11B/13B									
Baichuan2-Chat-13B [39]	56.63	82.07	68.04	77.71	44.44	34.48	50.00	33.92	40.71
LLAMA 2-CHAT-13B [35]	41.52	65.89	47.42	60.14	55.56	51.72	64.43	62.19	58.48
Vicuna-13B [37]	58.80	82.89	69.85	78.83	62.96	41.38	73.71	44.52	55.64
FLAN-T5-11B [43]	78.91	88.94	79.64	86.05	59.26	48.28	72.16	70.67	62.59
LLMs API									
GPT-3.5 [33]	77.72	94.76	89.43	93.10	64.81	65.52	75.77	71.02	69.28
GPT-4 [34]	86.52	95.34	91.24	94.07	75.93	68.97	79.38	72.44	74.29
Finetuned Models									
BERT <sub>large</sub> [6]	73.15	52.62	31.96	46.19	87.04	75.86	78.87	80.57	80.59
RoBERTa <sub>large</sub> [7]	73.26 <sup>◇</sup>	57.04	35.05	50.20	88.89	75.86	81.96	81.63	82.09
DeBERTa <sub>large</sub> [8]	85.65 <sup>◇</sup>	81.61	65.72	76.66	90.74	79.31	82.47	83.39	83.98
TEAM-DeBERTa [45]	89.35 <sup>◇</sup>	88.82	78.87	85.73	88.89	79.31	82.99	83.04	83.56
DILR-DeBERTa [46] <sup>†</sup>	47.66	59.14	47.68	55.57	88.89	82.76	84.54	83.75	84.99
NeuNLI-DeBERTa [28] <sup>†</sup>	88.15	89.87	79.12	86.53	90.74	82.76	84.02	83.04	85.14
FINAL(Ours)	90.33	91.97	81.70	88.77	94.44	86.21	85.05	85.51	87.80

The best results achieved by open source models are marked in bold. The best results across all models are marked in underline. <sup>◊</sup> denotes the results from Ghosal et al. [45].  
<sup>†</sup>: The original papers [28], [46] do not report the results of DeBERTa<sub>large</sub>, so we use the official public code to perform a grid search on important hyperparameters for the best results.

### A. Main Results

When comparing NeuNLI [28] with FINAL, it becomes apparent that our method performs better, with improvements of 2.18% on QASC, 2.24% on WorldTree, and 2.66% on WikiHop, as shown in Table VI. This improvement is credited to our utilization of first-order logic to decompose the hypothesis, which proves beneficial for questions requiring the composition of multiple sentences. Table VI also highlights that our method outperforms the DILR [46] method by 2.81% on the WikiHop dataset. This demonstrates the superior performance of our method, which incorporates a natural logic module. This module aids in the simplification of unstructured natural language, resulting in improved alignment with the outcomes of first-order logic reasoning. In addition, we also notice that the DILR method underperforms on the QASC and WorldTree datasets, indicating its suitability for well-structured queries rather than natural language questions. In contrast, our method performs well across all three datasets. The consistent results on the WikiHop dataset further demonstrate the generalizability and robustness of our framework, potentially showing its adaptability to various data patterns.

Our method also surpasses the performance of TEAM [45], with improvements of 0.98% on QASC, 3.04% on WorldTree and 4.24% on WikiHop. Both approaches incorporate contextual semantic information with DeBERTa for question answering and treat the task as a binary classification task. However, our

inclusion of first-order logic and natural logic not only improves performance but also provides a reasonable explanation for the results obtained.

On the WorldTree dataset, our method achieves the highest accuracy on both difficulty levels compared to other open source models, with 91.97% for the Easy level and 81.70% for the Challenge level. This indicates exceptional capabilities in understanding and reasoning. It is worth noting that both GPT-3.5 and GPT-4 exhibit higher accuracy than other models, emphasizing the excellent performance of large language models in commonsense reasoning tasks. Notably, GPT-3.5 and GPT-4 achieve higher accuracy on the WorldTree dataset with 93.10% and 94.07%, respectively. Conversely, their performance is relatively weaker on the QASC dataset. We conjecture that this discrepancy can be explained by the unique characteristics of the QASC dataset, which features 4 adversarial choices. These adversarial choices introduce greater uncertainty in predictions, posing a challenge for LLMs as they often struggle with prediction between the top-2/3 choices. However, our proposed method achieves state-of-the-art performance on the QASC dataset, overcoming the challenge posed by the presence of adversarial choices.

### B. Ablation Study

To evaluate the impact of different modules, we conduct an ablation study on the test set of QASC and WorldTree, as well



TABLE VII  
ABLATION STUDY BASED ON THE QASC, WORLDTREE, AND WIKIHOP DATASETS

Model	QASC Accuracy (%)	$\Delta$	WorldTree Accuracy (%)				WikiHop Accuracy (%)					
			Easy	Challenge	Overall	$\Delta$	Publisher	Developer	Country	Record_Label	Average	$\Delta$
FINAL	90.33	-	91.97	81.70	88.77	-	94.44	86.21	85.05	85.51	87.80	-
FINAL(w/o first-order logic)	87.39	-2.94	88.82	79.90	86.05	-2.72	90.74	79.31	83.51	83.39	84.24	-3.56
FINAL(w/o natural logic)	87.07	-3.26	89.76	77.58	85.97	-2.80	88.89	82.76	84.02	82.69	84.59	-3.21
FINAL(word-level)	88.26	-2.07	90.34	79.64	87.01	-1.76	92.59	82.76	84.54	83.39	85.82	-1.98

“w/o” indicates the removal of the corresponding module from the model.  $\Delta$  indicates the % difference from FINAL. FINAL(word-level) refers to the variation that adopts word-level mutation in the natural logic module.

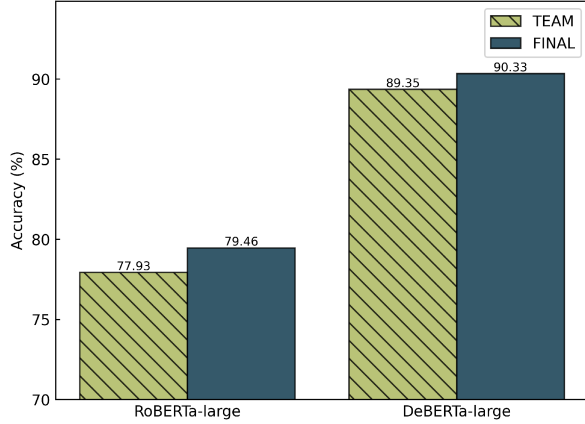


Fig. 2. Performance on the QASC dataset using different language models, RoBERTa-large and DeBERTa-large, for token representations.

as the dev set of WikiHop. The study consists of three variations of our model: 1) **FINAL(w/o first-order logic)**: This variation only uses the natural logic module for reasoning, without incorporating the first-order logic module. 2) **FINAL(w/o natural logic)**: This variation only uses the first-order logic module for reasoning, excluding the natural logic module. 3) **FINAL(word-level)**: This variation retains both the first-order logic module and the natural logic module for reasoning. However, in the natural logic module, word-level mutation is used instead of phrase-level mutation. The experimental results are presented in Table VII.

*Effectiveness of the First-order Logic Based Module:* Comparing FINAL with FINAL(w/o first-order logic), we notice a significant decrease in performance. The average accuracy score on the WikiHop development set drops from 87.80% to 84.24%. The same trend can be observed in the other two datasets. This indicates that certain questions require the composition of multiple sentences. The alignment with the triples obtained from the first-order logic module serves as a guiding force, ensuring that the inferred information is not only semantically meaningful but also logically coherent.

*Effectiveness of the Natural Logic Based Module:* Comparing the performance of FINAL with and without the inclusion of the natural logic module, we find that the inclusion of this module significantly enhances performance (+3.26% on QASC, +2.80% on WorldTree and +3.21% on WikiHop). This finding highlights the valuable role of natural logic in facilitating the reasoning process, especially in simplifying complex and unstructured natural language into a more streamlined and easily processed natural

language. Our observations suggest that both the first-order logic module and the natural logic module play equally crucial roles in our model.

*Effectiveness of Phrase-level Natural Logic Reasoning:* When comparing FINAL with FINAL using word-level natural logic, a noticeable performance degradation on FINAL(word-level) is apparent (−2.07% on QASC, −1.76% on WorldTree and −1.98% on WikiHop). This indicates that leveraging phrase-level natural logic-based reasoning is beneficial. While phrase-level mutations can introduce noise and errors, the inherent pruning techniques in the natural logic module can effectively address these issues. The experimental results also demonstrate that the benefits of phrase-level replacements outweigh the potential drawbacks, ensuring that the original semantics of the text are preserved and overall performance is enhanced.

### C. Comparison of Different Language Models for Token Representations

The performance of different language models for token representations on the QASC dataset is presented in Fig. 2. For comparison, we include the results of the TEAM method, which achieved the second-best performance on this dataset. The experimental results consistently show that our method, which used either the DeBERTa or RoBERTa model as the encoder, consistently outperforms their respective TEAM methods. This suggests that the integration of symbolic methods contributes to performance enhancement.

To explore the potential of integrating large language models with our framework, we conduct experiments using LLAMA2 as the encoder [49]. The results of our experiments are summarized in Table VIII. The integration of LLAMA2 as the encoder resulted in significant performance improvements on the WorldTree and WikiHop datasets. However, a slight decrease in accuracy was observed on the QASC dataset. We attribute this decrease to the unique challenges presented by the QASC dataset, which involves up to 8 answer choices, making it more difficult for the model to distinguish the correct answer. Additionally, FINAL<sub>LLAMA2</sub> might achieve better performance on the QASC dataset if a larger training corpus were available for fine-tuning, considering its extensive parameters.

### D. Complexity Analysis

In this section, we provide an analysis of the time and space complexity of the proposed FINAL framework.

*Time Complexity Analysis:* Our method consists of several key components: first-order logic module, natural logic module, and

TABLE VIII  
PERFORMANCE OF FINAL INTEGRATING LLAMA2 AS THE ENCODER ACROSS QASC, WORLDTREE, AND WIKIHOP DATASETS

Model	QASC Accuracy (%)	WorldTree Accuracy (%)			WikiHop Accuracy (%)				
		Easy	Challenge	Overall	Publisher	Developer	Country	Record_Label	Average
FINAL(Ours)	<b>90.33</b>	91.97	81.70	88.77	94.44	86.21	85.05	85.51	87.80
FINAL <sub>LLAMA2</sub> (Ours)	89.78	91.97	83.25	<b>89.25</b>	94.44	89.66	85.57	87.28	<b>89.24</b>

FINAL<sub>LLAMA2</sub> refers to our framework utilizing LLAMA2 as the encoder for token representations.

semantic similarity calculation module. In the first-order logic module, for a rule with  $k$  atoms, the hypothesis is decomposed into  $k$  sub-hypotheses. The time complexity is  $O(k)$ . For each intermediate premise, the natural logic reasoning module processes each word in the sentence. In the worst-case scenario, if a sentence has  $n$  words and each word can be mutated, the time complexity is  $O(n)$ . The number of sub-hypotheses is  $k$ , and correspondingly we have  $k$  intermediate premises. Thus the total complexity of the natural logic module is  $O(k \cdot n)$ . In the semantic similarity calculation module, we need to matching each sub-hypothesis with the corresponding intermediate premise. For  $k$  sub-hypotheses, the time complexity for matching is  $O(k)$ . Combining these components, the total time complexity of our method is  $O(k) + O(k \cdot n) + O(k) = O(k \cdot n)$ .

*Space Complexity Analysis:* The space complexity of our method primarily depends on the storage requirements for the token vector representations and the generated intermediate premises in the natural logic module. Assuming each token is represented by a vector of size  $d$  and there are  $n$  tokens in each sequence, the space complexity for storing these representations is  $O(n \cdot d)$ . For  $k$  sub-hypotheses, the total space complexity for token representation is  $O(k \cdot n \cdot d)$ . The intermediate premises generated during natural logic reasoning need to be stored temporarily. The number of intermediate premises depends on the maximum search depth  $\lambda$  and the number of mutations per step. Assuming a branching factor  $b$ , the total number of intermediate premises is  $O(b^\lambda)$ . The space complexity for storing these intermediate results is  $O(b^\lambda \cdot n \cdot d)$ . Combining these components, the total space complexity of our method is  $O(k \cdot n \cdot d) + O(b^\lambda \cdot n \cdot d) = O((k + b^\lambda) \cdot n \cdot d)$ .

#### E. Influence of Numbers of Fact Pairs and Inference Steps

In order to analyze the impact of different hyperparameters on the performance of our model, we conduct a series of experiments on the development set of the WorldTree dataset. The results are presented in Figs. 3 and 4.

*Effect of Number of Fact Pairs ( $\mu$ ):* The accuracy of our model remains consistent at approximately 93% as the number of relevant fact pairs increases from 1 to 4, as shown in Fig. 3. This indicates that even just one relevant fact pair obtained from the retrieval module contains sufficient knowledge for reasoning. Moreover, our model demonstrates efficient utilization of a smaller amount of external knowledge. It is important to note that adding more external knowledge pairs does not significantly enhance accuracy. In fact, increasing the number of knowledge pairs may introduce redundant information rather than providing valuable insights.

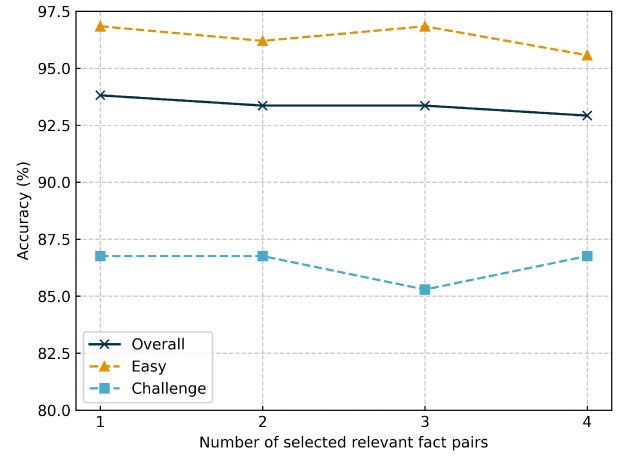


Fig. 3. Performance of FINAL under different numbers  $\mu$  of fact pairs.

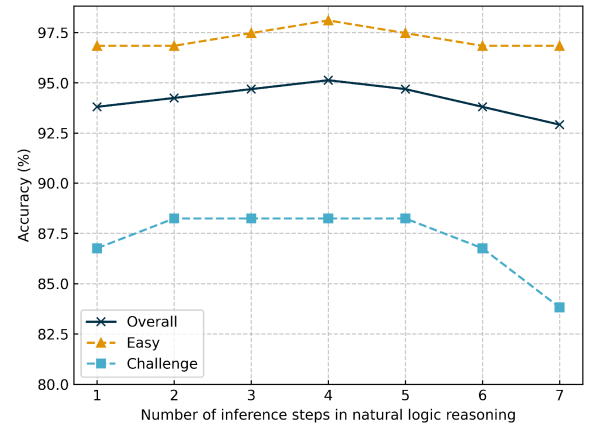


Fig. 4. Performance of FINAL under different numbers  $\lambda$  of reasoning steps.

*Effect of Number of Inference Steps ( $\lambda$ ) in Natural Logic Module:* Fig. 4 illustrates a clear enhancement in the accuracy of our model as the number of reasoning steps increases from 1 to 4. This can be attributed to the fact that the inferred sentences align more closely with the triplets obtained from the first-order logic module. However, it is important to note that beyond 4 reasoning steps, the accuracy begins to decline. This decline is a result of potential divergence from the original semantics caused by an increase in the number of reasoning steps, leading to what is known as semantic drift.

#### F. Distribution of Inference Steps

An analysis is conducted to determine the distribution of inference steps for the predicted state-of-the-art results on the

Percentage of Different Inference steps

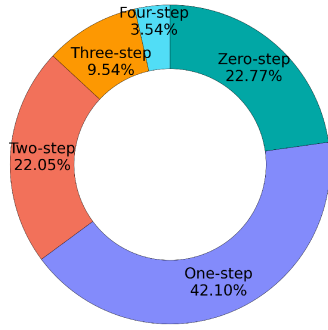


Fig. 5. Percentage distribution of various inference steps in the natural logic module.

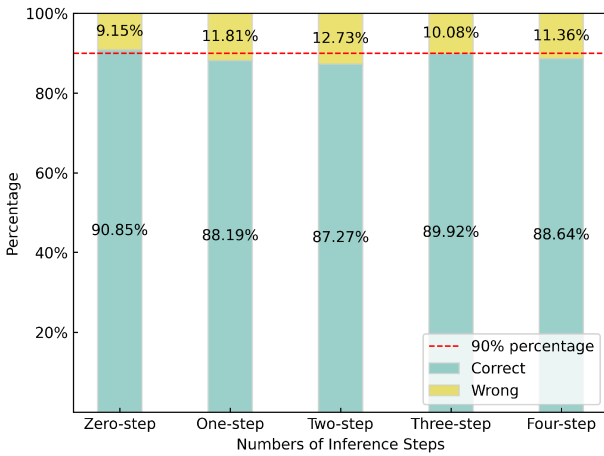


Fig. 6. Percentage of correctly predicted questions across different inference steps.

WorldTree dataset by FINAL. The distribution is visualized in Fig. 5. It shows that over three-quarters of the questions are predicted with at least one step, and more than a quarter of the questions are predicted with multiple steps. Notably, the threshold of four inference steps appears to be valid, providing sufficient reasoning for the task while avoiding unnecessary excessive inference steps. The majority of questions (42.10%) are impacted by 1-step inference. This indicates that, for a substantial portion of the test set, a single step of reasoning is sufficient for the model to make accurate predictions. Furthermore, an evaluation is performed on the percentage of correctly predicted questions across different inference steps, as illustrated in Fig. 6. It is observed that regardless of the number of inference steps, approximately 90% of questions are predicted correctly, emphasizing the efficacy of our method.

### G. Word-Level Versus Phrase-Level Mutation

In our analysis of the natural logic module, we examined the distribution of word-level and phrase-level mutations. The results are presented in Fig. 7. Our findings indicate that natural logical reasoning primarily takes place at the phrase level.

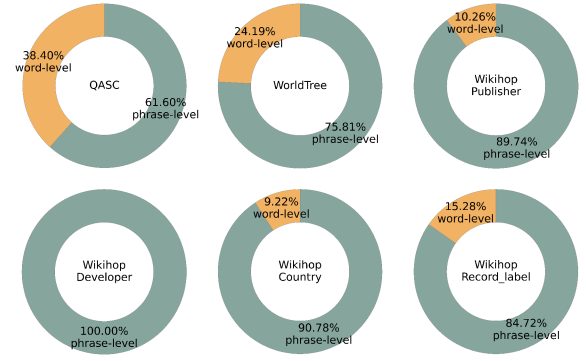


Fig. 7. Percentage of word-level reasoning versus phrase-level reasoning in the natural logic module.

TABLE IX  
THE FLEISS' KAPPA VALUE AND THE AVERAGE EXPLAINABILITY SCORE FROM HUMAN EVALUATIONS FOR NEUNLI AND FINAL OUTCOMES

	Fleiss' Kappa Value	Avg. Explainability Score
NeuNLI	0.59	0.36
FINAL	0.67	<b>0.55*</b>

\* denotes a significance test at the level of 0.05.

Specifically, it accounts for 61.60% on the QASC dataset, 75.81% on the WorldTree dataset, and nearly over 85% on the Wikihop dataset. These results suggest that, for natural logic, a key aspect of effectively understanding and reasoning about external knowledge involves focusing on information at the phrase level.

### H. Human Evaluation of Explainability

To evaluate the explainability of our model, we conduct a quantitative evaluation through human evaluations. For comparison, NeuNLI [28] is set as the baseline, and a significance test is conducted using the paired *t*-test at a significance level of 0.05. To ensure a fair comparison, we randomly selected 100 questions from the WorldTree dataset that were correctly classified by both NeuNLI and our model, FINAL.

The results of Fleiss' Kappa values and the average explainability scores are presented in Table IX, with the significance difference being less than 0.05. According to the commonly utilized interpretation criterion,<sup>2</sup> Fleiss' Kappa value for NeuNLI (0.59) achieves at least moderate agreement ( $\geq 0.41$ ) and the value for FINAL (0.67) reaches at least substantial agreement ( $\geq 0.61$ ). These results underscore the quality of the human annotations. The average explainability score of FINAL is significantly higher than that of NeuNLI and all annotators reach an agreement on the improvement by FINAL. This improvement can be ascribed to the ability of our method to break down complex hypotheses through the first-order module, as well as the mutation at the phrase level which enhances the inference process.

<sup>2</sup> [https://en.wikipedia.org/wiki/Fleiss%27\\_kappa](https://en.wikipedia.org/wiki/Fleiss%27_kappa)



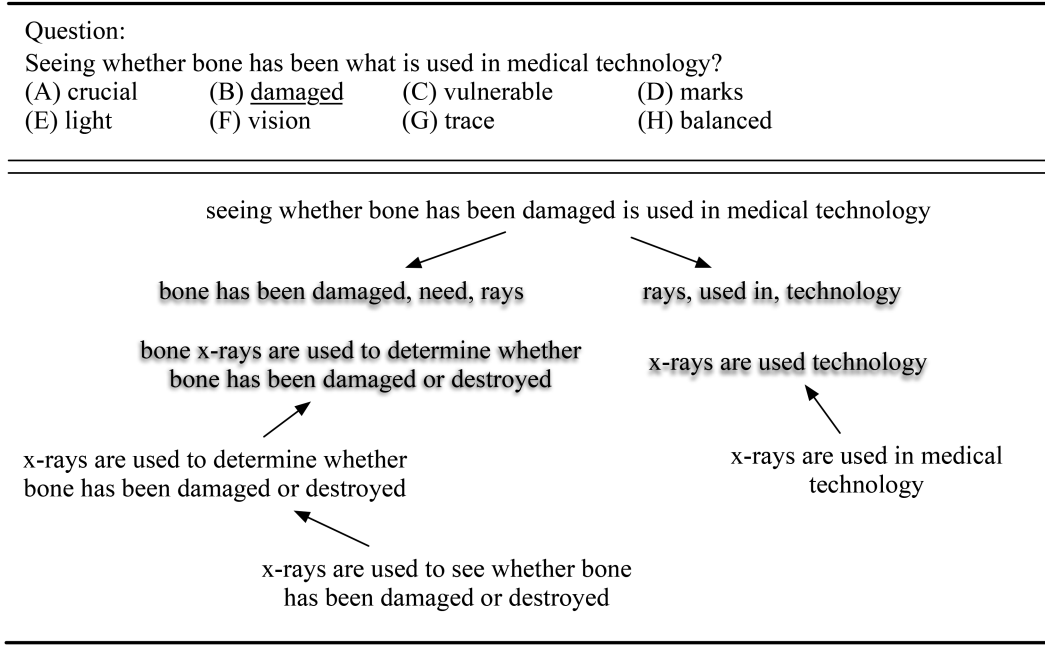


Fig. 8. Example of an explanation path generated by our model FINAL, showing a combination of first-order logic and natural logic. The correct answer is underlined.

TABLE X  
CASE STUDY FOR THE QASC DATASET

<p><i>Question:</i> some excellent what have jaws that have rows of teeth that grow through life? (A) frogs (B) backbones (C) arthropods (D) alligators (E) <u>predators</u> (F) evolution (G) bats (H) mice</p>
<p><b>FINAL</b> Predicated Answer: (E). ✓ First-order logic outputs: (some excellent predators, be, animals), (animals, have, jaws that have rows of teeth) Inferred fact<sub>1</sub>: jaws make cartilaginous animals excellent predators. Inferred fact<sub>2</sub>: cartilaginous animals have rows of teeth that grow throughout life.</p>
<p><b>GPT-3.5</b> Predicated Answer: (C). ✗ Reason: <i>Arthropods have jaws that have rows of teeth that grow throughout life</i>, making them the best choice for this question.</p>
<p><b>GPT-4</b> Predicated Answer: (D). ✗ Reason: <i>Alligators are known for having jaws with multiple rows of teeth that can grow throughout their lives</i>. When an alligator loses a tooth, another tooth grows in to replace it. This process can happen multiple times over the alligator's lifetime.</p>

The correct answer is underlined. Content marked in *italics* represents the hallucinatory output.

## I. Case Study

In this section, we provide an illustrative example to demonstrate the effectiveness of our method. The hypothesis presented in Fig. 8 states that “seeing whether bone has been damaged

is used in medical technology”. Upon processing through the first-order logic module, two tuples are derived: *[bone has been damaged, need, rays]* and *[rays, used in, technology]*. The shared argument between the two tuples is “rays”. Leveraging external knowledge, the retrieval module identifies the top candidate pair as (“x-rays are used to see whether bone has been damaged or destroyed”, “x-rays are used in medical technology”). Analyzing the reasoning paths, the left path shows the natural logic reasoning steps where FINAL transforms “see” into “determine” and appends “bone” before “x-rays” according to the context. On the other hand, the right reasoning path shows a step where the adjective “medical” is removed. Ultimately, the outputs from both logic modules convey similar semantic meanings and FINAL successfully identifies the key shared argument “rays”. In contrast, the result from GPT-3.5 is inaccurate. It predicts the answer “(A) crucial”, with the reasoning that “The word ‘crucial’ fits the context of determining the suitability of bone for medical technology, as it implies the importance and necessity of the assessment”. This indicates a misunderstanding of the context.

An example of incorrect predictions by GPT-3.5 and GPT4 in the QASC dataset is illustrated in Table X. The GPT4 predicts “alligators”, which is a hallucination. Although alligators have jaws, they do not have multiple rows of teeth as required by the question. The ground truth fact is that alligators grow new teeth, but they do not possess multiple rows of teeth.

## VI. RELATED WORK

The field of question answering (QA) has seen significant advancements over the years, evolving from traditional rule-based systems [50], [51] to advanced neural network models [33], [35].

Early QA systems [50], [51] relied heavily on symbolic and rule-based approaches. While effective for structured data, these methods struggled with the complexity and variability of natural language, leading to limited applicability in more diverse QA scenarios. The advent of machine learning introduced methods leveraging supervised learning and early neural networks, such as convolutional encoders [52] and recurrent encoders [53], which improved the handling of unstructured data by learning representations of text. Severyn et al. [54] demonstrated the effectiveness of multi-layered perceptrons combined with convolutional encoded representations for QA tasks. Despite these advances, these models often faced limitations in understanding context and managing long-range dependencies, which are critical for accurate QA.

The introduction of pre-trained language models (PLMs) such as BERT [6], RoBERTa [7], and DeBERTa [8] marked a substantial leap forward. These models significantly improved contextual language understanding through deep learning and contextual embeddings. Nevertheless, PLMs [55], [56] often lack interpretability, functioning as black boxes that provide little insight into their decision-making processes. This opacity can be problematic in applications requiring transparency and trust. Recent advancements with models such as ChatGPT [33], LLaMA 2 [35], and Gemma [41] have further advanced the field. These models employ techniques like cloze prompting [57] and multiple choice prompting [44] to achieve impressive performance on QA tasks. However, they also introduce challenges related to transparency. To address the interpretability issue, researchers have explored the generation of free-text explanations using large language models [58]. While these explanations aim to elucidate the model's reasoning, they often fail to accurately reflect the model's underlying beliefs and can suffer from hallucinations [59]. Structured explanations, on the other hand, offer a more transparent way to trace the model's decision-making process, enhancing our understanding of its behavior.

Neuro-symbolic approaches provide a promising direction in this regard. These methods [11], [60], [61], [62], [63], [64], [65] leverage the robust learning capabilities of neural networks while incorporating the structured reasoning processes of symbolic systems. For instance, Rocktäschel et al. [9], Weber et al. [10], and Minervini et al. [66] have explored integrating first-order logic and neural networks using differentiable unification operators to facilitate backward-chain reasoning. Furthermore, some research efforts [63], [64], [65] leverage language models to convert natural language problems into symbolic formulations, which are then solved using deterministic symbolic solvers. However, these methods often rely on synthetic language datasets that are easily parsed into first-order logic, limiting their applicability to more complex, real-world scenarios.

## VII. CONCLUSION

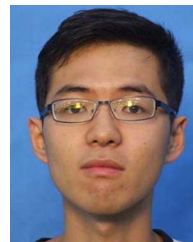
In this paper, we address question-answering tasks by framing them as textual entailment tasks. While current neural methods are effective, they often lack transparency in their decision-making processes. To address this limitation and provide clear insights into the rationale for correct decisions, we propose a

neuro-symbolic framework, called FINAL. Our proposed framework combines the systematic reasoning process of first-order logic with the flexibility of natural logic to enhance differentiable reasoning. This combination allows for the integration of unstructured external knowledge bases, which is a limitation in traditional first-order logic approaches. Specifically, we utilize first-order logic to decompose the hypothesis and natural logic to construct reasoning paths from the premise to the hypothesis. We employ bidirectional reasoning to establish links along the reasoning path, ensuring a transparent decision-making process. Extensive experiments conducted on three diverse datasets demonstrate the superior performance of our proposed framework compared to existing methods. Our framework not only achieves higher accuracy but also provides clearer insights into the decision-making process. Future work will continue to explore the integration of large language models to further enhance the reasoning capabilities and scalability of the framework, aiming to address even more complex question-answering tasks.

## REFERENCES

- [1] G. Angeli, N. Nayak, and C. D. Manning, "Combining natural logic and shallow reasoning for question answering," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2016, pp. 442–452.
- [2] Z. Yang et al., "HotpotQA: A dataset for diverse, explainable multi-hop question answering," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 2369–2380.
- [3] D. Khashabi, S. Chaturvedi, M. Roth, S. Upadhyay, and D. Roth, "Looking beyond the surface: A challenge set for reading comprehension over multiple sentences," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2018, pp. 252–262.
- [4] D. Dua, Y. Wang, P. Dasigi, G. Stanovsky, S. Singh, and M. Gardner, "DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2019, pp. 2368–2378.
- [5] T. Khot, P. Clark, M. Guerquin, P. Jansen, and A. Sabharwal, "QASC: A dataset for question answering via sentence composition," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 8082–8090.
- [6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2019, pp. 4171–4186.
- [7] Y. Liu et al., "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.
- [8] P. He, J. Gao, and W. Chen, "Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing," in *Proc. Int. Conf. Learn. Representations*, 2022, pp. 1–16.
- [9] T. Rocktäschel and S. Riedel, "End-to-end differentiable proving," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017.
- [10] L. Weber, P. Minervini, J. Münchmeyer, U. Leser, and T. Rocktäschel, "NLProlog: Reasoning with weak unification for question answering in natural language," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2019.
- [11] A. Smirnova et al., "Nessy: A neuro-symbolic system for label noise reduction," *IEEE Trans Knowl Data Eng*, 2022.
- [12] G. Lakoff, "Linguistics and natural logic," *Synthese*, 1970.
- [13] B. MacCartney and C. D. Manning, "An extended model of natural logic," in *Proc. 8th Int. Conf. Comput. Semantics*, 2009, pp. 140–156.
- [14] J. Van Benthem, "Determiners and logic," *Linguistics Philosophy*, vol. 6, no. 4, pp. 447–478, 1983.
- [15] T. F. Icard III and L. S. Moss, "Recent progress on monotonicity," *Linguistic Issues Lang. Technol.*, vol. 9, pp. 167–194, 2014.
- [16] P. Clark et al., "Think you have solved question answering? try ARC, the A12 reasoning challenge," 2018, *arXiv:1803.05457*.
- [17] A. Horn, "On sentences which are true of direct unions of algebras 1," *J. Symbolic Log.*, vol. 16, pp. 14–21, 1951.
- [18] C. Lin, A. Chaudhury, A. B. Whinston, and D. C. Marinescu, "Logical inference of horn clauses in petri net models," *IEEE Trans Knowl Data Eng*, vol. 5, no. 3, pp. 416–425, Jun. 1993.

- [19] J. Grant, J. Gryz, J. Minker, and L. Raschid, "Logic-based query optimization for object databases," *IEEE Trans Knowl Data Eng*, vol. 12, no. 4, pp. 529–547, Jul./Aug. 2000.
- [20] S. Schoenmackers, J. Davis, O. Etzioni, and D. Weld, "Learning first-order horn clauses from web text," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2010, pp. 1088–1098.
- [21] W. Y. Wang and W. W. Cohen, "Learning first-order logic embeddings via matrix factorization," in *Proc. Int. Joint Conf. Artif. Intell.*, 2016, pp. 2132–2138.
- [22] S. Yang, R. Zhang, S. Erfani, and J. H. Lau, "An interpretable neuro-symbolic reasoning framework for task-oriented dialogue generation," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 4918–4935.
- [23] V. M. S. Valencia, "Studies on natural logic and categorial grammar," Universiteit van Amsterdam, 1991. [Online]. Available: <https://eprints.illc.uva.nl/id/eprint/1849/2/HDS-17-Victor-Sanchez.text.pdf>
- [24] C. Gormley and Z. Tong, *Elasticsearch: The Definitive Guide*. Philadelphia, PA, USA: O'Reilly Media, Inc., 2015.
- [25] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd, "spaCy: Industrial-strength natural language processing in python," 2020. [Online]. Available: <https://spacy.io/>
- [26] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–13.
- [27] M. Lewis et al., "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 7871–7880.
- [28] J. Shi, X. Ding, L. Du, T. Liu, and B. Qin, "Neural natural logic inference for interpretable question answering," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2021, pp. 3673–3684.
- [29] G. A. Miller, "WordNet: A lexical database for english," *Commun. ACM*, vol. 38, pp. 39–41, 1995.
- [30] P. Jansen, E. Wainwright, S. Marmorstein, and C. Morrison, "WorldTree: A corpus of explanation graphs for elementary science questions supporting multi-hop inference," in *Proc. Lang. Resour. Eval. Conf.*, 2018, pp. 2732–2740.
- [31] J. Welbl, P. Stenetorp, and S. Riedel, "Constructing datasets for multi-hop reading comprehension across documents," *Trans. Assoc. Comput. Linguistics*, vol. 6, pp. 287–302, 2018.
- [32] J. L. Fleiss, B. Levin, and M. C. Paik, *Statistical Methods for Rates and Proportions*. Hoboken, NJ, USA: Wiley, 2013.
- [33] OpenAI, "Introducing chatgpt," 2022. [Online]. Available: <https://openai.com/index/chatgpt/>
- [34] J. Achiam et al., "GPT-4 technical report," 2023, *arXiv:2303.08774*.
- [35] H. Touvron et al., "Llama 2: Open foundation and fine-tuned chat models," 2023, *arXiv:2307.09288*.
- [36] AI, Meta, "Llama 3 model card," 2024. [Online]. Available: [https://github.com/meta-llama/llama3/blob/main/MODEL\\_CARD.md](https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md)
- [37] W.-L. Chiang et al., "Vicuna: An open-source chatbot impressing GPT-4 with 90% chatgpt quality," 2023. [Online]. Available: <https://lmsys.org/blog/2023-03-30-vicuna/>
- [38] Z. Du et al., "GLM: General language model pretraining with autoregressive blank infilling," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 320–335.
- [39] Baichuan, "Baichuan 2: Open large-scale language models," 2023, *arXiv:2309.10305*.
- [40] A. Q. Jiang et al., "Mistral 7B," 2023, *arXiv:2310.06825*.
- [41] G. Team et al., "Gemma: Open models based on gemini research and technology," 2024, *arXiv:2403.08295*.
- [42] J. Bai et al., "Qwen technical report," 2023, *arXiv:2309.16609*.
- [43] H. W. Chung et al., "Scaling instruction-finetuned language models," *J. Mach. Learn. Res.*, vol. 25, pp. 1–53, 2024.
- [44] J. Robinson and D. Wingate, "Leveraging large language models for multiple choice question answering," in *Proc. Int. Conf. Learn. Representations*, 2023, pp. 1–28.
- [45] D. Ghosal, N. Majumder, R. Mihalcea, and S. Poria, "Two is better than many? binary classification as an effective approach to multi-choice question answering," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2022, pp. 10158–10166.
- [46] W. Wang and S. Pan, "Deep inductive logic reasoning for multi-hop reading comprehension," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 4999–5009.
- [47] T. Wolf et al., "Transformers: State-of-the-Art natural language processing," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2020, pp. 38–45.
- [48] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–18.
- [49] P. BehnamGhader, V. Adlakha, M. Mosbach, D. Bahdanau, N. Chapados, and S. Reddy, "LLM2Vec: Large language models are secretly powerful text encoders," 2024, *arXiv:2404.05961*.
- [50] M. Yahya, K. Berberich, S. Elbassuoni, M. Ramanath, V. Tresp, and G. Weikum, "Natural language questions for the web of data," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2012, pp. 379–390.
- [51] Q. Cai and A. Yates, "Large-scale semantic parsing via schema matching and Lexicon extension," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2013, pp. 423–433.
- [52] X. Zhang, S. Li, L. Sha, and H. Wang, "Attentive interactive neural networks for answer selection in community question answering," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 3525–3531.
- [53] Y. Tay et al., "Learning to rank question answer pairs with holographic dual LSTM architecture," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2017, pp. 695–704.
- [54] A. Severyn and A. Moschitti, "Learning to rank short text pairs with convolutional deep neural networks," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2015, pp. 373–382.
- [55] W. Yang et al., "End-to-end open-domain question answering with bert-serini," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2019, pp. 72–77.
- [56] C. Raffel et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, pp. 5485–5551, 2020.
- [57] F. Petroni et al., "Language models as knowledge bases?," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2019, pp. 1772–1791.
- [58] F. Huang, H. Kwak, and J. An, "Chain of explanation: New prompting method to generate quality natural language explanation for implicit hate speech," in *Proc. Int. Conf. World Wide Web*, 2023, pp. 90–93.
- [59] L. Huang et al., "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions," 2023, *arXiv:2311.05232*.
- [60] H. Dong, J. Mao, T. Lin, C. Wang, L. Li, and D. Zhou, "Neural logic machines," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–22.
- [61] R. Manhaeve et al., "DeepProbLog: Neural probabilistic logic programming," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 3753–3763.
- [62] E. van Krieken, E. Acar, and F. van Harmelen, "Analyzing differentiable fuzzy logic operators," *Artif. Intell.*, vol. 302, 2022, Art. no. 103602.
- [63] L. Pan, A. Albalak, X. Wang, and W. Wang, "Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning," in *Proc. Findings Empirical Methods Natural Lang. Process.*, 2023, pp. 3806–3824.
- [64] X. Ye, Q. Chen, I. Dillig, and G. Durrett, "SATLM: Satisfiability-aided language models using declarative prompting," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2023, pp. 45548–45580.
- [65] T. X. Olausson et al., "LINC: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2023, pp. 5153–5176.
- [66] P. Minervini, M. Bosnjak, T. Rocktäschel, S. Riedel, and E. Grefenstette, "Differentiable reasoning on large knowledge bases and natural language," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 5182–5190.



**Jihao Shi** (Student Member, IEEE) is currently working toward the PhD degree with the school of computer science and technology, Harbin Institute of Technology, China. He is a visiting PhD degree with the school of computer science and engineering, Nanyang Technological University, Singapore, which is funded by the China Scholarship Council. His main research interests include natural language processing, interpretable commonsense reasoning, and question answering.



**Xiao Ding** received the PhD degree in computer science and technology from the Harbin Institute of Technology, Harbin, China, in 2016. Currently, he is a professor with the Harbin Institute of Technology. His current research interests include natural language processing, social computing and text mining.

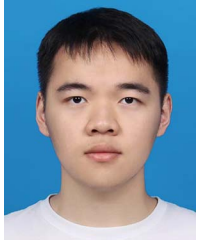




**Siu Cheung Hui** received the BSc degree in mathematics and the DPhil degree from the University of Sussex, Brighton, U.K., in 1983 and 1987, respectively. He is currently an associate professor with the College of Computing and Data Science, Nanyang Technological University, Singapore. His current research interests include information retrieval, natural language processing, and semantic web.



**Ting Liu** received the PhD degree from the Department of Computer Science, Harbin Institute of Technology, China, in 1998. He is currently a full professor in the Department of Computer Science, Harbin Institute of Technology. His research interests include information retrieval, natural language processing, and social media analysis.



**Yuxiong Yan** is currently the student with the Harbin Institute of Technology. His current research interests are in natural language processing, eventic graph, and commonsense reasoning.



**Hengwei Zhao** is currently the student with the Harbin Institute of Technology. His current research interests are in natural language processing, eventic graph and question answering.



**Bing Qin** received the PhD degree from the Department of Computer Science, Harbin Institute of Technology, China, in 2005. She is currently a full professor in the Department of Computer Science, Harbin Institute of Technology. Her research interests include natural language processing, information extraction, document-level discourse analysis, and sentiment analysis.