Logic-Driven Context Extension and Data Augmentation for Logical Reasoning of Text

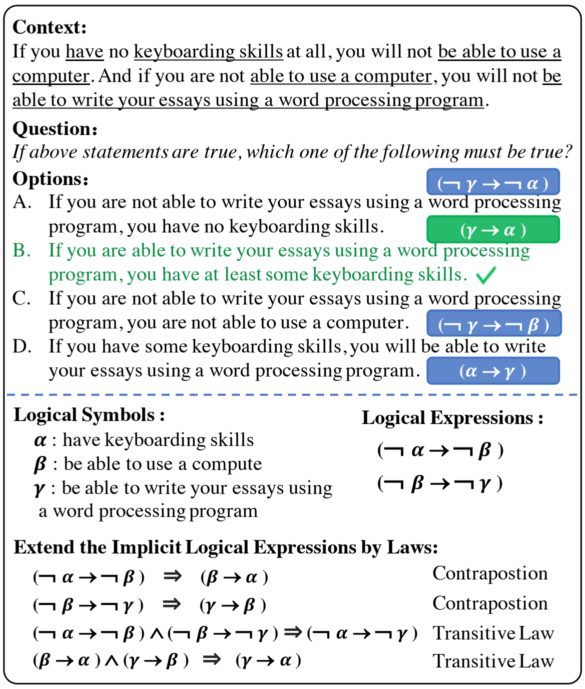
Siyuan Wang[1]*, **Wanjun Zhong**[2], **Duyu Tang**[3], **Zhongyu Wei**[1,5],
Zhihao Fan[1], **Daxin Jiang**[3], **Ming Zhou**[4], **Nan Duan**[3]
[1]School of Data Science, Fudan University, China
[2]Sun Yat-Sen University, China [3]Microsoft, China [4]Sinovation Ventures, China
[5]Research Institute of Intelligent and Complex Systems, Fudan University, China
{wangsy18,zywei,fanzh18}@fudan.edu.cn; zhongwj25@mail2.sysu.edu.cn
zhouming@chuangxin.com; {dutang,djiang,nanduan}@microsoft.com

Abstract

Logical reasoning of text requires identifying critical logical structures in the text and performing inference over them. Existing methods for logical reasoning mainly focus on contextual semantics of text while struggling to explicitly model the logical inference process. In this paper, we not only put forward a logic-driven context extension framework but also propose a logic-driven data augmentation algorithm. The former follows a three-step reasoning paradigm, and each step is respectively to extract logical expressions as elementary reasoning units, symbolically infer the implicit expressions following equivalence laws and extend the context to validate the options. The latter augments literally similar but logically different instances and incorporates contrastive learning to better capture logical information, especially logical negative and conditional relationships. We conduct experiments on two benchmark datasets, ReClor and LogiQA. The results show that our method achieves state-of-the-art performance on both datasets, and even surpasses human performance on the ReClor dataset. [1]

1 Introduction

Recent years have witnessed a growing interest in logical reasoning of text, which learns to understand a given text in logical level and perform logical inference to deduce implications from asserted ones (McCarthy, 1989; Nilsson, 1991). As a significant component of human reading comprehension, it is essential in many application scenarios, such as negotiation and debate. And several datasets have been proposed as benchmarks for this task (Williams et al., 2017; Habernal et al., 2017; Yu et al., 2020; Liu et al., 2020).


Context:
If you have no keyboarding skills at all, you will not be able to use a computer. And if you are not able to use a computer, you will not be able to write your essays using a word processing program.

Question:
If above statements are true, which one of the following must be true?

Options:
A. If you are not able to write your essays using a word processing program, you have no keyboarding skills. $(\neg\gamma \rightarrow \neg\alpha)$
B. If you are able to write your essays using a word processing program, you have at least some keyboarding skills. ✓ $(\gamma \rightarrow \alpha)$
C. If you are not able to write your essays using a word processing program, you are not able to use a computer. $(\neg\gamma \rightarrow \neg\beta)$
D. If you have some keyboarding skills, you will be able to write your essays using a word processing program. $(\alpha \rightarrow \gamma)$

Logical Symbols :
$\boldsymbol{\alpha}$: have keyboarding skills
$\boldsymbol{\beta}$: be able to use a compute
$\boldsymbol{\gamma}$: be able to write your essays using a word processing program

Logical Expressions :
$(\neg\boldsymbol{\alpha} \rightarrow \neg\boldsymbol{\beta})$
$(\neg\boldsymbol{\beta} \rightarrow \neg\boldsymbol{\gamma})$

Extend the Implicit Logical Expressions by Laws:
$(\neg\alpha \rightarrow \neg\beta) \Rightarrow (\beta \rightarrow \alpha)$ Contrapostion
$(\neg\beta \rightarrow \neg\gamma) \Rightarrow (\gamma \rightarrow \beta)$ Contrapostion
$(\neg\alpha \rightarrow \neg\beta) \wedge (\neg\beta \rightarrow \neg\gamma) \Rightarrow (\neg\alpha \rightarrow \neg\gamma)$ Transitive Law
$(\beta \rightarrow \alpha) \wedge (\gamma \rightarrow \beta) \Rightarrow (\gamma \rightarrow \alpha)$ Transitive Law

Figure 1: A logical reasoning example from ReClor dataset (Yu et al., 2020). To find the answer, it needs to extract *logical symbols*, identify *logical expressions* and perform logical inference to extend the *implicit logical expressions*. The underlined phrases represent logical symbols. The colored rectangles are corresponding logical expressions of each option.

An example of logical reasoning problems is shown in Figure 1, which takes a context description, a question and four options as the input, and aims to identify the option that logically follows the context. The main challenge to solve such a problem is to uncover the logical propositional structure among the text and perform logical inference over them, which are beyond the capability of contextual pre-trained models (Liu et al., 2019; Yang et al., 2019; Lan et al., 2020) without such logical annotations. They usually treat logical reasoning as a traditional reading comprehension task and match the given context with candidate answers, without modeling the discrete logical inference process explicitly (Yu et al., 2020). Recently, Huang et al. (2021) utilizes discourse information to unwrap

[*] Work is done during internship at Microsoft. Zhongyu Wei and Duyu Tang are corresponding authors.

[1] Codes are publicly available at https://github.com/WangsyGit/LReasoner.

Findings of the Association for Computational Linguistics: ACL 2022, pages 1619 - 1629
May 22-27, 2022 ©2022 Association for Computational Linguistics

the logical structure and propose a discourse-aware graph network to learn discourse-based contextual embeddings for logical reasoning. However, it is still entangled in enhancing contextual representation while ignoring explicit logical inference.

In responding to these issues, we propose a three-step paradigm for logical reasoning based on symbolic logic information. Firstly, we identify the elementary components for reasoning from the context as the logical expressions, like $(\neg\alpha \to \neg\beta)$, to uncover the logical relationships between logical symbols. Then we perform logical inference following equivalence laws to extend the implicit ones from these identified logical expressions. Thirdly, candidate options can be validated by comparing themselves with all obtained logical expressions.

We propose a logic-driven context extension framework to integrate these three reasoning steps, namely logic identification to parse the context into logical expressions, logic extension to infer implicit logical expressions and logic verbalization for answer prediction. To combine the interpretability of symbolic inference with anti-noise of continuous representation, we follow a neural-symbolic paradigm (Besold et al., 2017; Garcez et al., 2019) which conducts logic identification and extension in a symbolic manner and utilizes the pre-trained model as the backbone of logic verbalization. In practice, we verbalize implicit logical expressions into natural language and feed them as an extended context into a pre-trained model to match the answer. Moreover, to encourage the pre-trained model to better capture logical information, we further propose a logic-driven data augmentation algorithm. Specifically, it constructs challenging instances with literally similar but logically different contexts by modifying logical expressions. Contrastive learning (Chen et al., 2020) is used for encouraging our model to distinguish different contexts to better capture negative and conditional relationships in logical expressions.

The experiments are conducted on two challenging logical reasoning datasets, ReClor (Yu et al., 2020) and LogiQA (Liu et al., 2020). Results show that our system achieves state-of-the-art performance on both datasets, and even surpasses human performance on ReClor. Further results also show the effectiveness of both logic-driven context extension framework and data augmentation algorithm, and demonstrate the generalizability of our system.

2 Task and Background

2.1 Task Definition

We study the problem of logical reasoning of text on a multiple-choice question answering task. The task is described as following: given a context c, a question q, and four associated options $\{o_1, o_2, o_3, o_4\}$, we aim to select the most appropriate option as the answer o_a.

2.2 Base Model

In this paper, we follow the leading methods on the leaderboards to take pre-trained models as our base model, e.g., RoBERTa (Liu et al., 2019). It concatenates the context, the question and each option as an input and encodes the sequence for calculating its score. Given four options, four concatenated sequences are constructed to calculate four scores, and the one with the highest score is chosen as the answer. Specifically, the concatenated sequence is formulated as $[CLS]\ c\ [SEP]\ q\ ||\ o\ [SEP]$, where c is the context and $q\ ||\ o$ is the concatenation of the question and each option. The representations of special token $[CLS]$ in four sequences are fed into a linear layer with a softmax function to get the probability distribution of options as $P(\{o_1, o_2, o_3, o_4\}|c, q)$. The cross entropy loss is calculated as Eq. 1, where o_a is the correct option.

$$\mathcal{L}_A = -\sum \log P(o_a|c, q) \tag{1}$$

Although promising results have been reported (Yu et al., 2020), pre-trained models for logical reasoning directly encode the triplet of context, question and options, which mainly leverage contextual semantics but struggle to model the symbolic inference process explicitly. Thus we propose a framework on top of a pre-trained model to extract logical expressions in the text and symbolically perform logical inference to predict the answer.

3 Logic-Driven Context Extension

In this section, we present a logic-driven context extension framework for logical reasoning of text, which is illustrated in Figure 2. The framework is divided into three steps as follows. It first identifies the logical symbols and expressions explicitly mentioned in the context and options (§ 3.1). Then it performs interpretable logical inference over them to extend the logical expressions implicit in the context (§ 3.2). Finally, it verbalizes the extended logical expressions related to each option as an

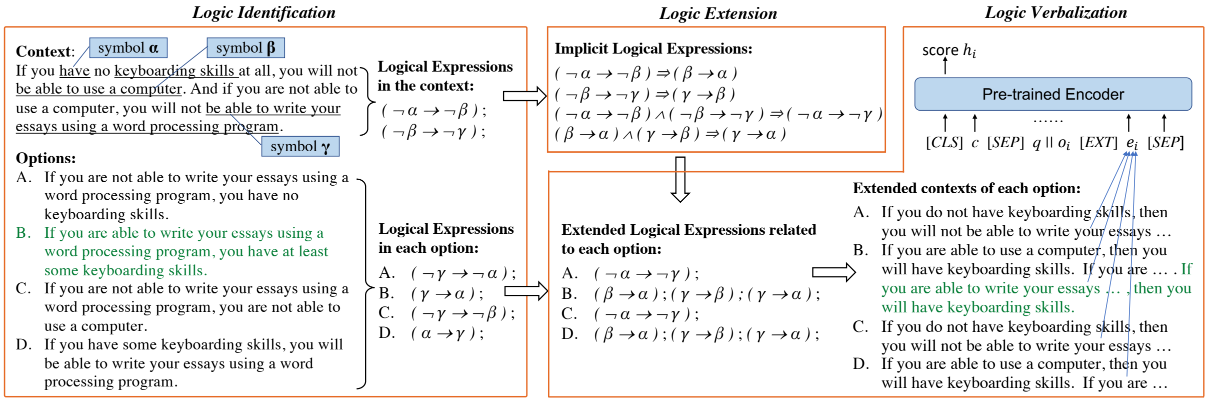

Logic Identification
Context:
symbol α
symbol β
symbol γ
If you have no keyboarding skills at all, you will not be able to use a computer. And if you are not able to use a computer, you will not be able to write your essays using a word processing program.
Options:
A. If you are not able to write your essays using a word processing program, you have no keyboarding skills.
B. If you are able to write your essays using a word processing program, you have at least some keyboarding skills.
C. If you are not able to write your essays using a word processing program, you are not able to use a computer.
D. If you have some keyboarding skills, you will be able to write your essays using a word processing program.
Logical Expressions in the context:
(¬α → ¬β) ;
(¬β → ¬γ) ;
Logical Expressions in each option:
A. (¬γ → ¬α) ;
B. (γ → α) ;
C. (¬γ → ¬β) ;
D. (α → γ) ;
Logic Extension
Implicit Logical Expressions:
(¬α → ¬β) ⇒ (β → α)
(¬β → ¬γ) ⇒ (γ → β)
(¬α → ¬β) ∧ (¬β → ¬γ) ⇒ (¬α → ¬γ)
(β → α) ∧ (γ → β) ⇒ (γ → α)
Extended Logical Expressions related to each option:
A. (¬α → ¬γ) ;
B. (β → α) ; (γ → β) ; (γ → α) ;
C. (¬α → ¬γ) ;
D. (β → α) ; (γ → β) ; (γ → α) ;
Logic Verbalization
score h_i
Pre-trained Encoder
......
[CLS] c [SEP] q || o_i [EXT] e_i [SEP]
Extended contexts of each option:
A. If you do not have keyboarding skills, then you will not be able to write your essays …
B. If you are able to use a computer, then you will have keyboarding skills. If you are … . If you are able to write your essays … , then you will have keyboarding skills.
C. If you do not have keyboarding skills, then you will not be able to write your essays …
D. If you are able to use a computer, then you will have keyboarding skills. If you are …

Figure 2: The overall architecture of logic-driven context extension framework. c, q, o_i and e_i are the context, question, i-th option and the extended context for i-th option, respectively. The texts in green mean that the option B is matched against its extended context which has the highest score.

extended context and utilizes it in the pre-trained model to match the answer (§ 3.3).

3.1 Logic Identification

In order to perform logical reasoning, we first need to identify the elementary reasoning components as logical expressions to uncover the logical relationships between logical symbols. We identify the existing logical expressions for each sentence in the context and each option. To show the format of the logical expression, we introduce some notations:

(1) $\{\alpha, \beta, \gamma, ...\}$: the logical symbols, which are the basic constituents in the context to constitute the logical expressions, such as the "have keyboarding skills" in Figure 2.

(2) $\{\neg, \rightarrow\}$: the logical connective set. $\neg$ means the negation operation upon a specific logical symbol and $\rightarrow$ acts as a conditional relationship between two logical symbols.

(3) $\{(\alpha \rightarrow \beta), ...\}$: the logical expressions, which are composed of logical symbols and connectives. $(\alpha \rightarrow \beta)$ means that α is the condition of β.

To ensure the generalizability of our framework without annotated logic forms, we design a fairly simple logical identification approach using an off-the-shelf constituency parser (Joshi et al., 2018) and several common keywords of logical semantics. We first employ the constituency parser to extract constituents including noun phrases and gerundial phrases as basic symbols. The logical symbols in each sentence are combined by logical connectives to constitute logical expressions as follow-up. If any negative word (e.g., "*not*", "*unable*") is in or immediately before a logical symbol α, we add the negation connective $\neg$ before α as a new symbol $\neg\ \alpha$. Then if there is a conditional relationship between two symbols α and β in a sentence, we construct the corresponding logical expression as $(\alpha \rightarrow \beta)$. We simply recognize the conditional relationship between α and β as $(\alpha \rightarrow \beta)$ according to conditional indicators (e.g., *"if α, then β"*, *"β since α"*) and whether an active voice occurs between α and β. The detailed negative and conditional keywords are listed in Appendix A with the whole identification procedure summarized as an algorithm. As shown in Figure 2, given the context with two sentences, we can extract three logical symbols $\{\alpha, \beta, \gamma\}$ and identify two existing logical expressions as $(\neg\alpha \rightarrow \neg\beta)$ and $(\neg\beta \rightarrow \neg\gamma)$.

3.2 Logic Extension

In addition to the logical expressions explicitly mentioned in the context, there are still some other implicit ones that we need to logically infer and extend. We combine the identified logical expressions existing in all sentences of the context as a logical expression set $\mathcal{S}$, and perform logical inference over them to further extend the implicit expressions according to logical equivalence laws. Here we follow two most applicable logical equivalence laws involving implication and negation in propositional logic, including *contraposition* (Russel et al., 2013) and *transitive law* (Zhao et al., 1997):

Contraposition :

$$(\alpha \rightarrow \beta) \implies (\neg\beta \rightarrow \neg\alpha) \tag{2}$$

Transitive Law :

$$(\alpha \rightarrow \beta) \wedge (\beta \rightarrow \gamma) \implies (\alpha \rightarrow \gamma) \tag{3}$$

Then the extended implicit logical expressions form an extension set of the current logical expression set $\mathcal{S}$ as $\mathcal{S}_E$. As in Figure 2, the set of existing logical expressions is $\mathcal{S} = \{(\neg\alpha \rightarrow \neg\beta), (\neg\beta \rightarrow \neg\gamma)\}$ and the logic extension set is $\mathcal{S}_E = \{(\beta \rightarrow \alpha), (\gamma \rightarrow \beta), (\neg\alpha \rightarrow \neg\gamma), (\gamma \rightarrow \alpha)\}$.

3.3 Logic Verbalization

After inferring the extended logical expression set $\mathcal{S}_E$, we verbalize them into natural language for better utilization of the pre-trained model considering that symbolic logic is more difficult to be encoded. We first select the related expressions from $\mathcal{S}_E$ for each option. A logical expression is regarded as related to an option if it has the same logical symbols with the option judged by the text overlapping and whether a negation connective exists. For example, $(\neg\alpha \rightarrow \neg\gamma)$ in Figure 2 is related to option C because they both contain $\neg\gamma$. Then we transform all logical expressions related to the option at symbolic space into natural language by filling them into a template and concatenate them into a sentence. We take such a sentence as an extended context for this option. For simplicity, we only adopt the If-Then statements as the verbalization template, which is one of the most common patterns of logical reasoning, but we make some adjustments according to the tense and singular/plural. Specifically, the template is designed as shown in Table 1.

Logic	$(\neg\alpha \rightarrow \neg\gamma)$
Template	If do not α, then will not γ.
Extended context	If you do not have keyboarding skills, then you will not be able to write your essays using a word processing program.

Table 1: An example of verbalizing a logical expression into text.

We feed extended contexts into the pre-trained model to match the options and predict the answer. We take an extended context as the sentence e, and introduce a special token $[EXT]$ to represent context extension. Then we reformulate the input sequence as $[CLS]\ c\ [SEP]\ q\ ||\ o\ [EXT]\ e\ [SEP]$ for encoding and feed the $[CLS]$ representation into a classification layer to get each option's score and find the most appropriate answer.

4 Logic-Driven Data Augmentation

In order to make the pre-trained model put more focus on logical information in the context, especially logical negative and conditional relationships, we further introduce a logic-driven data augmentation algorithm. Inspired by SimCLR (Chen et al., 2020), we augment challenging instances with literally similar but logically different contexts built by modifying logical expressions. It then adopts contrastive learning and encourages our model to distinguish logically correct context supporting the answer. We first introduce the background of SimCLR and then describe our logic-driven contrastive learning.

SimCLR As a paradigm of self-supervised representation learning by comparing different samples, contrastive learning (Wu et al., 2018; He et al., 2020a) aims to make the representations of similar samples be mapped close together, while that of dissimilar samples be further away in the encoding space. The goal can be described as following.

$$s(f(x), f(x^+)) \gg s(f(x), f(x^-)) \quad (4)$$

x^+ is a positive sample similar to the data point x while x^- is a negative sample dissimilar to x. $f(\cdot)$ is an encoder to learn a representation and the $s(\cdot)$ is a similarity function of two representations. Over this, SimCLR (Chen et al., 2020) builds a classifier to distinguish positive from negative samples and learns to capture what makes two samples different.

Logic-Driven Contrastive Learning In our question answering setting, we alter the score function from measuring the similarity between two representations towards calculating the score that the question can be solved by the correct answer under a given context:

$$s^{'}(c^+, q, o_a) \gg s^{'}(c^-, q, o_a) \quad (5)$$

where (c^+, q, o_a) and (c^-, q, o_a) are the positive and negative sample, c^+ and c^- are the positive and negative context, respectively, and $s^{'}$ is the score function. The contrastive loss can be formulated as a classification loss for predicting the most plausible context that supports the answer:

$$\mathcal{L}_C = -\sum \log \frac{\exp(s^{'}(+))}{\exp(s^{'}(+)) + \exp(s^{'}(-))} \quad (6)$$

where $s^{'}(+)$ and $s^{'}(-)$ are short for $s^{'}(c^+, q, o_a)$ and $s^{'}(c^-, q, o_a)$ respectively.

Aware of symbolic logical expressions, we can construct *logical negative samples* including negative contexts that are literally similar but logical dissimilar to the positive one. We take the original context to construct the positive sample. Then we generate a negative sample by modifying the existing logical expressions in the context and verbalizing the modified logical expressions into a negative context as § 3.3. During the modification operations, we randomly choose a logical expression and randomly *delete, reverse or negate* such an expression. The *delete, reverse or negate* operations are respectively to delete a logical expression in the context, reverse the conditional order of a logical expression and negate a logical symbol in a logical expression. The constructing procedure of a logical negative sample is illustrated in Figure 3. Then the model can be trained to better capture logical information, especially negative and conditional relationships in logical expressions.

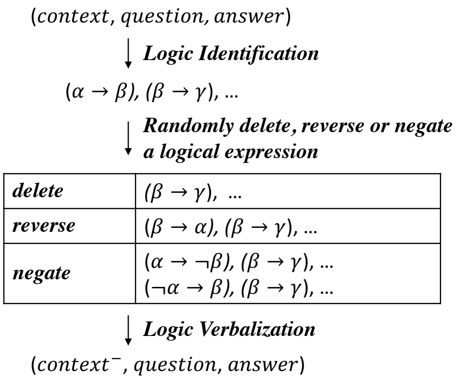


Figure 3: Procedure to construct a logical negative sample.

In the logic-driven data augmentation algorithm, our framework is trained with a combined loss as $\mathcal{L} = \mathcal{L}_A + \mathcal{L}_C$. And the classification of positive and negative context for the correct answer is also implemented in the logic-driven context extension framework.

5 Experiments

5.1 Experimental Dataset

Our experiments are conducted on two challenging datasets ReClor (Yu et al., 2020) and LogiQA (Liu et al., 2020) that cover diverse and complicated logical reasoning skills, to investigate the general effectiveness of our system. ReClor is built upon standardized exams including GMAT and LSAT. As there are some biased instances that can be solved without knowing contexts and questions, ReClor splits the unbiased instances from the test data as the HARD set to fully assess the logical reasoning ability. The other biased ones are taken as the EASY set. LogiQA comes from the National Civil Servants Examination of China and is professionally translated into an English version.

ReClor consists of $6,138$ questions and is divided into training, validation and test sets with $4,638$, 500 and $1,000$ data points. The test set is further split into EASY set and HARD set with 440 and 560 data points. LogiQA contains $8,678$ questions and is split into $7,376/651/651$ samples for training, validation and testing. Each question is collected with a context and four answer options, in which only one is correct. The implementation details of experiments are given in Appendix B.

5.2 Overall Performance

We compare our systems with several baseline models and human performance.

Baseline Models The compared baseline pre-trained models include *BERT* (Devlin et al., 2019), *RoBERTa* (Liu et al., 2019), *ALBERT* (Lan et al., 2020) and *DeBERTa* (He et al., 2020b). We also compare our model with *DAGN* (Huang et al., 2021), an available state-of-the-art method on the leaderboard which proposes a discourse-aware graph network for logical reasoning taking RoBERTa-large as the backbone.

Our Systems $LReasoner_{RoBERTa}$ is our proposed logic-driven **reasoner** taking RoBERTa as the backbone model, which utilizes both logic-driven context extension framework and data augmentation algorithm. We also build our *LReasoner* on top of two more powerful pre-trained models ALBERT and DeBERTa as $LReasoner_{ALBERT}$ and $LReasoner_{DeBERTa}$, respectively. Besides, $LReasoner_{Ensemble}$ is an ensemble of *DeBERTa*, $LReasoner_{ALBERT}$ and $LReasoner_{DeBERTa}$.

Human Performance Yu et al. (2020) and Liu et al. (2020) report human performance as the average scores of graduate or post-graduate students over randomly chosen test samples.

The evaluation results are shown in Table 2. We have several findings:

- Our systems outperform all baseline models on both datasets by a considerable margin. $LReasoner_{Ensemble}$ even surpasses the human performance on both EASY and HARD sets of ReClor. This indicates the effectiveness of our method for logical reasoning.
- Compared to the corresponding baseline models including *RoBERTa*, *ALBERT* and *DeBERTa*, our $LReasoner_{RoBERTa}$, $LReasoner_{ALBERT}$ and

Model	ReClor				LogiQA	
	Val	Test	EASY	HARD	Val	Test
BERT (Devlin et al., 2019)*	53.8	49.8	72.0	32.3	33.8	32.1
RoBERTa (Liu et al., 2019)*	62.6	55.6	75.5	40.0	35.9	35.3
ALBERT (Lan et al., 2020)	70.2	66.5	76.6	58.6	38.9	37.6
DeBERTa (He et al., 2020b)	74.4	68.9	83.4	57.5	44.4	41.5
DAGN (Huang et al., 2021)	65.8	58.3	75.9	44.5	36.9	39.3
LReasoner$_{RoBERTa}$	66.2	62.4	81.4	47.5	38.1	40.6
LReasoner$_{ALBERT}$	73.2	70.7	81.1	62.5	41.6	41.2
LReasoner$_{DeBERTa}$	**74.6**	**71.8**	**83.4**	**62.7**	**45.8**	**43.3**
LReasoner$_{Ensemble}$	78.0	76.1	87.0	67.5	45.8	45.0
Human Performance*	-	63.0	57.1	67.2	-	86.0

Table 2: Experimental results (accuracy %) of different models on ReClor and LogiQA. The results in **bold** are the best performance of each column except for *LReasoner*$_{Ensemble}$ and *Human Performance*. * indicates that the results of ReClor and LogicQA are taken from (Yu et al., 2020) and (Liu et al., 2020).

LReasoner$_{DeBERTa}$ consistently perform better. It demonstrates that our method is robust to be effective for logical reasoning based on different pre-trained models, even the most recent state-of-the-art ones.

- Our models generate large improvement on both HARD and EASY sets of ReClor compared with baseline models. This observation verifies that our model is capable of improving logical reasoning ability on both biased and unbiased data.

5.3 Further Analysis

Ablation Study To dive into the effectiveness of different components in our logic-driven reasoner, we conduct an ablation study which takes *RoBERTa* as our backbone model on ReClor validation and test sets. As shown in Table 3, *RoBERTa+CE* and *RoBERTa+DA* both outperform the baseline model *RoBERTa* and perform worse than our final system *RoBERTa+CE+DA*. It indicates that both logic-driven context extension framework and data augmentation algorithm can boost the performance of question answering involving logical reasoning.

Model	Val	Test	EASY	HARD
RoBERTa	62.6	55.6	75.5	40.0
+ *CE*	65.2	58.3	78.6	42.3
+ *DA*	65.8	61.0	80.9	45.4
+ *CE* + *DA*	66.2	62.4	81.4	47.5

Table 3: Ablation study of our system. *CE* and *DA* are respectively our logic-driven context extension framework and data augmentation algorithm. *RoBERTa+CE+DA* is our proposed *LReasoner*$_{RoBERTa}$.

Comparison of Negative Sample Construction Strategies To further analyze the effectiveness of our logical negative samples in logic-driven contrastive learning, we compare several different negative sample construction strategies in contrastive learning on top of *RoBERTa* for ReClor.

Model	Test	EASY	HARD
RoBERTa (w/o CLR)	55.6	75.5	40.0
RoBERTa (w/ CLR-RS)	58.2	79.3	41.6
RoBERTa (w/ CLR-RD)	58.9	78.9	43.2
RoBERTa (w/ CLR-L)	61.0	80.9	45.4

Table 4: Comparison of different negative sample construction approaches. *CLR* represents contrastive learning. *RS* means randomly selecting a context from in-batch data while *RD* means randomly deleting a sentence from the original context. *L* denotes our logical negative sample construction method in logic-driven contrastive learning.

From Table 4, we can find that all models with contrastive learning outperform the model without it, which demonstrates that contrastive learning can help to better predict the answer. Our logic-driven contrastive learning *RoBERTa(w/ CLR-L)* performs best. It reveals that logical negative samples are more effective than negative samples constructed by other methods which make the model better capture the logical negative and conditional relationships in the context for logical reasoning.

Evaluation of Logic Identification To evaluate the performance of our symbolic logic identification method, we randomly sample 50 instances from the validation set and manually annotate the logical symbols and expressions as labels. We re-

Context : Everyone sitting in the clubhouse of the golf course today at ten o' clock had just registered for a beginner' s golf lesson. Gerald, Robert, and Shirley were sitting in the clubhouse this morning at ten o' clock. No accomplished golfer would register for a beginner' s golf lesson.

Question : If the statements above are true, which one of the following must also be true on the basis of them?

Options : (Answer : C)

A. Gerald, Robert, and Shirley were the **only** people who registered for a beginner 's golf lesson this morning. ($\gamma \rightarrow$ Others)

B. None of the people sitting in the clubhouse this morning at ten o' clock **had ever played golf**. ($\alpha \rightarrow \neg$ Others)

C. Neither Gerald nor Shirley is an accomplished golfer. ($\gamma \rightarrow \neg \eta$)

D. Everyone sitting in the clubhouse this morning at ten o' clock registered **only** for a beginner's golf lesson. ($\alpha \rightarrow$ Others)

Logical Symbols & Expressions	α : sitting in the clubhouse of the golf course today at ten o‘ clock; β : registered for a beginner‘ s golf lesson ; γ : Gerald, Robert, and Shirley; η: accomplished golfer ; $\alpha \rightarrow \beta$; $\gamma \rightarrow \alpha$; $\eta \rightarrow \neg \beta$;
Extending the Implicit Logical Expressions	$(\alpha \rightarrow \beta) \Rightarrow (\neg \beta \rightarrow \neg \alpha)$; $(\gamma \rightarrow \alpha) \Rightarrow (\neg \alpha \rightarrow \neg \gamma)$; $(\eta \rightarrow \neg \beta) \Rightarrow (\beta \rightarrow \neg \eta)$; $(\alpha \rightarrow \beta) \wedge (\gamma \rightarrow \alpha) \Rightarrow (\gamma \rightarrow \beta)$; $(\neg \beta \rightarrow \neg \alpha) \wedge (\neg \alpha \rightarrow \neg \gamma) \Rightarrow (\neg \beta \rightarrow \neg \gamma)$; $(\alpha \rightarrow \beta) \wedge (\beta \rightarrow \neg \eta) \Rightarrow (\alpha \rightarrow \neg \eta)$; $(\eta \rightarrow \neg \beta) \wedge (\neg \beta \rightarrow \neg \alpha) \Rightarrow (\eta \rightarrow \neg \alpha)$; $(\gamma \rightarrow \beta) \wedge (\beta \rightarrow \neg \eta) \Rightarrow (\gamma \rightarrow \neg \eta)$; $(\eta \rightarrow \neg \alpha) \wedge (\neg \alpha \rightarrow \neg \gamma) \Rightarrow (\eta \rightarrow \neg \gamma)$;
Implicit Logical Expressions related to each option	A. $(\gamma \rightarrow \beta)$; $(\gamma \rightarrow \neg \eta)$; B. $(\alpha \rightarrow \neg \eta)$; C. $(\gamma \rightarrow \beta)$; $(\gamma \rightarrow \neg \eta)$; ✓ D. $(\alpha \rightarrow \neg \eta)$;

Figure 4: A ReClor case of the reasoning process of *LReasoner*$_{ALBERT}$. Phrases underlined denote other symbols (called *Others*) different from the logical symbols in context and **bold** tokens make them different.

port the recall of logical symbol and logical expression identification as 65.9% and 48.9%, respectively. We can see that our generic logic parsing method which operates in an unsupervised manner achieves relatively reliable performance. Unsupervised and generic logic parsing is an essential future direction that is expected to be further studied to enhance the performance of the overall system.

Case Study A ReClor case is presented in Figure 4 to show the reasoning process of our system. At first, the logical symbols are correctly extracted from the context and the logical expressions are identified based on them considering logical negative and conditional relationships. Then we extend the logical expressions by inferring implicit ones in the context. For each option, we recognize its logical expression and find the related extended expressions. We verbalize them into the text to feed into the pre-trained model as an extended context to compute a matching score. Finally, we take option C which exactly matches an extended implicit logical expression as the most plausible answer.

Detailed Analysis of Different Reasoning Types As ReClor integrates various types of logical reasoning skills, we can detailedly investigate the performance of our system *LReasoner*$_{ALBERT}$ on different logical reasoning types compared to the baseline model *ALBERT*. We analyze the improvements brought by our system, and point out challenges to shed a light on future directions.

As shown in Table 5, our model is generally effective on most reasoning types compared to the baseline model, especially `Implication`, `Most Strongly Supported`. These questions emphasize the ability of inference over logical units. Specifically, `Implication` needs to infer the conclusion that logically follows a set of premises while `Most Strongly Supported` aims to find the statement that is most strongly supported by a stimulus. This observation verifies the effectiveness of our system to model logical deduction. Besides, `Implication` is precisely the reasoning ability investigated by NLI tasks, which reveals that our model would also be effective in NLI.

However, there still exists some reasoning types

Reasoning Type	*Base*	*Ours*
Necessary Assumptions (11.0%)	73.7	76.3 (↑)
Sufficient Assumptions (3.6%)	70.0	70.0 (−)
Strengthen (9.0%)	69.1	70.2 (↑)
Weaken (10.6%)	64.6	59.3 (↓)
Evaluation (1.6%)	69.2	69.2 (−)
Implication (6.2%)	43.8	54.3 (↑)
Conclusion/Main Point (3.1%)	80.6	77.8 (↓)
Most Strongly Supported (6.7%)	58.9	71.4 (↑)
Explain or Resolve (8.0%)	60.7	67.9 (↑)
Principle (5.7%)	72.3	76.9 (↑)
Dispute (2.5%)	63.3	80.0 (↑)
Technique (3.8%)	75.0	80.6 (↑)
Role (3.7%)	78.1	68.8 (↓)
Identify a Flaw (11.3%)	65.0	71.8 (↑)
Match Flaws (4.9%)	61.3	61.3 (−)
Match the Structure (2.7%)	56.7	86.7 (↑)
Others (5.5%)	68.5	72.6 (↑)

Table 5: Results of different reasoning types. Numbers in parentheses are percentages of different types. *Base* is the *ALBERT* while *Ours* means our *LReasoner*$_{ALBERT}$. ↑, ↓ and − respectively mean that our performance is better, worse than and equal to the baseline *ALBERT*.

that are challenging for our system, such as Match flaws and Weaken. Weaken aims to find the opposite statement that weakens the argument. Match flaws is even more challenging as it requires analyzing the flaw that conflicts with the complete logical chain in the context, and finding an option exhibiting the same flaw. Therefore, how to model the different degrees of a logical statement, and abstract the complete logical chain for flaw identification, are interesting future directions.

5.4 Generalizability Discussion

Our logic-driven reasoner not only embodies its superiority in ReClor and LogiQA, but also can be generalized to other datasets and task formats. To demonstrate this, we evaluate our framework on a widely studied extractive QA task SQuAD (Rajpurkar et al., 2016), which covers diverse skills instead of just explicit logical reasoning, such as reasoning of lexical variation, commonsense and causal relations (Sugawara and Aizawa, 2016). As shown in Table 6, our framework is effective on SQuAD compared to both RoBERTa-base and RoBERTa-large, which manifests the generalizability of our logic-driven reasoner.

Model	EM	F1
*RoBERTa-base**	83.0	90.4
LReasoner$_{\textit{RoBERTa-base}}$	85.6	91.7
*RoBERTa-large**	88.9	94.6
LReasoner$_{\textit{RoBERTa-large}}$	89.3	94.8

Table 6: Dev. set results of our framework compared to RoBERTa (both base and large models) on SQuAD. * denotes the results come from (Liu et al., 2019).

6 Related Work

In recent years, there has been a surge in NLP research towards complex reasoning, such as reasoning for commonsense knowledge (Huang et al., 2019), numerical calculation (Dua et al., 2019) or multi-hop aggregation (Yang et al., 2018). Compare to these widely studied reasoning tasks, logical reasoning is also an essential and challenging capability but is relatively unexplored. Natural Language Inference (NLI) (Dagan et al., 2005; Bowman et al., 2015; Williams et al., 2018; Khot et al., 2018) is a typical task requiring logical reasoning, which aims to determine whether a hypothesis can be reasonably entailed from a premise. However, these NLI datasets mainly handle the task at sentence-level and are limited to only a few logical reasoning types, such as entailment, contradiction, and neutral. To promote a deeper passage-level logical reasoning ability, several QA datasets have been proposed. LogiQA (Liu et al., 2020) is collected from the National Civil Servants Examination of China covering 5 logical reasoning types. Yu et al. (2020) propose ReClor dataset from the GMAT and LSAT tests which examines 17 types of logical reasoning. In this paper, we take both ReClor and LogiQA as the testbed to investigate diverse and complicated logical reasoning skills.

Pre-trained language models (Devlin et al., 2019; Liu et al., 2019; Yang et al., 2019; Lan et al., 2020) have been widely adopted for various reasoning tasks and achieve promising performance. However, they directly encode the given texts to predict the output while failing to identify the symbolic logical structure and perform explicit logical inference for logical reasoning of text. Semantic parsers (Reddy et al., 2016; Singh et al., 2020) are usually employed for converting texts to logical forms, and graph neural networks (Fang et al., 2019; Huang et al., 2021) and neural module networks (Gupta et al., 2019) also have been attempted to partly imitate the human reasoning process. But these neural methods may not be easily generalized to our desired propositional logical schema without annotations and still perform an implicit inference. To circumvent these limitations and utilize the superior performance of neural models, we take inspiration from neuro-symbolic reasoning (Wang et al., 2018; Arabshahi et al., 2020) to integrate symbolic inference and neural representation. We design an explicit three-step logical reasoning paradigm and propose a logic-driven reasoning system to generically identify the logical structure and perform interpretable logical inference in a symbolic module while taking a pre-trained model as the backbone.

7 Conclusion and Future Work

In this paper, we focus on the task of logical reasoning of text. Following a three-step logical reasoning paradigm, we first propose a neuro-symbolic logic-driven context extension framework. It identifies logical expressions as elementary units of logical inference and symbolically deduces the implicitly mentioned expressions, and verbalizes them as an extended context into a pre-trained model to match the answer. We also introduce a logic-driven data augmentation algorithm, which augments literally similar but logically different instances and employs contrastive learning to help our model better capture logical information. Experimental results confirm the general effectiveness of our LReasoner, and it even surpasses human performance on the ReClor dataset. In the future, we will explore to model different logical reasoning types and directly incorporate symbolic logic into the model structure.

Acknowledgments

This work is partially supported by Natural Science Foundation of China (No.6217020551, 61906176), Science and Technology Commission of Shanghai Municipality Grant (No.20dz1200600, 21QA1400600, GWV-1.1, 21511101000) and Zhejiang Lab (No. 2019KD0AD01).

References

Forough Arabshahi, Jennifer Lee, Mikayla Gawarecki, Kathryn Mazaitis, Amos Azaria, and Tom Mitchell. 2020. Conversational neuro-symbolic commonsense reasoning. *arXiv preprint arXiv:2006.10022*.

Tarek R Besold, Artur d'Avila Garcez, Sebastian Bader, Howard Bowman, Pedro Domingos, Pascal Hitzler, Kai-Uwe Kühnberger, Luis C Lamb, Daniel Lowd, Priscila Machado Vieira Lima, et al. 2017. Neural-symbolic learning and reasoning: A survey and interpretation. *arXiv preprint arXiv:1711.03902*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, pages 177–190. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. *arXiv preprint arXiv:1903.00161*.

Yuwei Fang, Siqi Sun, Zhe Gan, Rohit Pillai, Shuohang Wang, and Jingjing Liu. 2019. Hierarchical graph network for multi-hop question answering. *arXiv preprint arXiv:1911.03631*.

Artur d'Avila Garcez, Marco Gori, Luis C Lamb, Luciano Serafini, Michael Spranger, and Son N Tran. 2019. Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning. *arXiv preprint arXiv:1905.06088*.

Nitish Gupta, Kevin Lin, Dan Roth, Sameer Singh, and Matt Gardner. 2019. Neural module networks for reasoning over text. *arXiv preprint arXiv:1912.04971*.

Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. 2017. The argument reasoning comprehension task: Identification and reconstruction of implicit warrants. *arXiv preprint arXiv:1708.01425*.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020a. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020b. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos qa: Machine reading comprehension with contextual commonsense reasoning. *arXiv preprint arXiv:1909.00277*.

Yinya Huang, Meng Fang, Yu Cao, Liwei Wang, and Xiaodan Liang. 2021. Dagn: Discourse-aware graph network for logical reasoning. *arXiv preprint arXiv:2103.14349*.

Vidur Joshi, Matthew Peters, and Mark Hopkins. 2018. Extending a parser to distant domains using a few dozen partially annotated examples. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1190–1199, Melbourne, Australia. Association for Computational Linguistics.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. Scitail: A textual entailment dataset from science question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*.

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. *CoRR*, abs/1907.11692.

I. Loshchilov and F. Hutter. 2017. Fixing weight decay regularization in adam. *ArXiv*, abs/1711.05101.

John McCarthy. 1989. Artificial intelligence, logic and formalizing common sense. In *Philosophical logic and artificial intelligence*, pages 161–190. Springer.

Nils J Nilsson. 1991. Logic and artificial intelligence. *Artificial intelligence*, 47(1-3):31–56.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas. Association for Computational Linguistics.

Siva Reddy, Oscar Täckström, Michael Collins, Tom Kwiatkowski, Dipanjan Das, Mark Steedman, and Mirella Lapata. 2016. Transforming dependency structures to logical forms for semantic parsing. *Transactions of the Association for Computational Linguistics*, 4:127–140.

Stuart Russel, Peter Norvig, et al. 2013. *Artificial intelligence: a modern approach*. Pearson Education Limited.

Hrituraj Singh, Milan Aggrawal, and Balaji Krishnamurthy. 2020. Exploring neural models for parsing natural language into first-order logic. *arXiv preprint arXiv:2002.06544*.

Saku Sugawara and Akiko Aizawa. 2016. An analysis of prerequisite skills for reading comprehension. In *Proceedings of the Workshop on Uphill Battles in Language Processing: Scaling Early Achievements to Robust Methods*, pages 1–5, Austin, TX. Association for Computational Linguistics.

Lei Wang, Yan Wang, Deng Cai, Dongxiang Zhang, and Xiaojiang Liu. 2018. Translating a math word problem to an expression tree. *arXiv preprint arXiv:1811.05632*.

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, New Orleans, Louisiana. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Weihao Yu, Zihang Jiang, Yanfei Dong, and Jiashi Feng. 2020. Reclor: A reading comprehension dataset requiring logical reasoning. In *International Conference on Learning Representations (ICLR)*.

J-K Zhao, Elizabeth M Rudnick, and Janak H Patel. 1997. Static logic implication with application to redundancy identification. In *Proceedings. 15th IEEE VLSI Test Symposium (Cat. No. 97TB100125)*, pages 288–293. IEEE.

A Details of Logic Identification

We design a generic logic identification approach that uses an off-the-shelf constituency parser and most common keywords of logical semantics (totally no more than 20). We employ the constituency parser to extract constituents as basic symbols. We regard literally similar constituents with an overlap rate over 60% as the same symbol if they also have consistent degree modifiers, such as "*only*", "*most*", "*least*", etc.

We define a set of negative words for identifying logical negation, including {"*not*", "*n't*", "*unable*", "*no*", "*few*", "*little*", "*neither*", "*none of*"}. And the full set of conditional indicators for recognizing the logical conditional relationship between α and β as $(\alpha \rightarrow \beta)$ is {"*if* α*, then* β", "α *in order for* β", "α *thus* β", "β *due to* α", "β *owing to* α", "β *since* α", "$\neg\beta$ *unless* α"}. The detailed parsing procedure is illustrated in Algorithm 1.

Algorithm 1 Logic Identification Algorithm

```
Input: A sentence in the context or an option $t$ to be parsed, a set of logical negative keywords $\mathcal{N}$ and a set of logical conditional indicators $\mathcal{C}$.
Output: A logical expressions set $\mathcal{S}$ parsed from the input $t$.
Initializing $\mathcal{S} := \{\}$
Extracting constituents from the input $t$.
Recognizing literally similar constituents as the same symbol and obtain all logical symbols as $\{\alpha, \beta, ...\}$.
for symbol $a$ in $\{\alpha, \beta, ...\}$ do
    if $\exists\, n_i \in \mathcal{N}$ is in or immediately before the logical symbol $a$ then
        Adding the negation connective $\neg$ before $a$ as $\neg a$.
        Replacing the original symbol with the negative one as $a := \neg a$.
    end if
end for
for symbol $a$ in $\{\alpha, \beta, ...\}$ do
    for symbol $b$ in $\{\alpha, \beta, ...\}$ do
        if $a \neq b$ and ( $\exists\, c_i \in \mathcal{C}$ is between two logical symbols $a$ and $b$ or an active voice occurs between $a$ and $b$ ) then
            Obtaining a logical expression $a \rightarrow b$.
            Appending $a \rightarrow b$ to the logical expression set $\mathcal{S}$.
        end if
    end for
end for
return The logical expressions set $\mathcal{S}$.
```

B Implementation Details

We take RoBERTa-large (Liu et al., 2019), ALBERT-xxlarge-v2 (Lan et al., 2020) and DeBERTa-xlarge (He et al., 2020b) as our backbones and implement them using Huggingface (Wolf et al., 2019). We use a batch size of 8 and fine-tune for 10 epochs. The AdamW (Loshchilov and Hutter, 2017) with $\beta 1$ = 0.9 and $\beta 2$ = 0.98 is taken as the optimizer and the learning rate is 1e-5. We use a linear learning rate scheduler with 10% warmup proportion. We automatically evaluate our model on validation set to choose parameters that achieve the highest accuracy. We select at most two extended logical expressions related to each option to construct the extended context for ReClor and select at most three for LogiQA. We train our proposed systems and other comparison models on two NVIDIA Tesla V100 GPUs.