Beta-LR: Interpretable Logical Reasoning based on Beta Distribution

Yizhuo Ma [1]
myz@std.uestc.edu.cn

Ke Qin [1] *
qinke@uestc.edu.cn

Shuang Liang [1,2] *
shuangliang@uestc.edu.cn

[1] Laboratory of Intelligent Collaborative Computing
University of Electronic Science and Technology of China, Chengdu, 611731, China
[2] Sichuan Artificial Intelligence Research Institute, Yibin, 644000, China

Abstract

The logical information contained in text is of significant importance for logical reasoning. Previous approaches have relied on embedding text into a low-dimensional vector to capture logical information and perform reasoning in Euclidean space. These methods involve constructing special graph architectures that match logical relations or designing data augmentation frameworks by extending texts based on symbolic logic. However, it presents two obvious problems. 1) The logical information reflected in the text exhibits uncertainty that is difficult to represent using a vector. 2) Integrating logical information requires modeling logical operations (such as $\cup$, $\cap$, and $\neg$), while only simple arithmetic operations can be performed in Euclidean space. To address both the problems, we propose Beta-LR, a probabilistic embedding method to capture logical information. Specifically, we embed texts into beta distribution on each dimension to eliminate logical uncertainty. We also define neural operators that enable interpretability and perform logical operations based on the characteristics of the beta distribution. We conduct experiments on two datasets, ReClor and LogiQA, and our Beta-LR achieves competitive results. The experiments demonstrate that our method effectively captures the logical information in text for reasoning purposes. The source code is available at https://github.com/myz12138/Beta-LR.

Context:
(s_1) The television show Henry was not widely watched until it was scheduled for Tuesday evenings immediately after That' s Life, (s_2) the most popular show on television. (s_3) During the year after the move, (s_4) Henry was consistently one of the ten most-watched shows on television. (s_5) Since Henry' s recent move to Wednesday evenings, (s_6) however, (s_7) it has been watched by far fewer people. (s_8) We must conclude that Henry was widely watched before the move to Wednesday evenings because it followed That' s Life and not because people especially liked it.

Question:
Which one of the following, if true, most strengthens the argument?

Options:
A. The show that now follows That's Life on Tuesdays has double the number of viewers it had before being moved.
B. Henry has been on the air for three years, but That's Life has been on the air for only two years.
C. After its recent move to Wednesday, Henry was aired at the same time as the second most popular show on television.
D. That's Life was not widely watched during the first year it was aired.

Answer: A

Figure 1: A logical reasoning example from ReClor (Yu et al.) dataset. It requires to learn logical information contained in sentences $\{s_1, s_2, ..., s_8\}$ of context and perform logical operations over them to integrate logical information, which will be used for answering the question.

1 Introduction

In recent years, there has been an increasing focus on logical reasoning (Nilsson, 1991; Habernal et al.; Liu et al., 2023), which presents a significant challenge as it necessitates the extraction of crucial information from text. Figure 1 provides an example of a logic reasoning problem taken from the ReClor (Yu et al.) dataset, which serves as a benchmark for evaluating logical reasoning capabilities. This dataset is in the form of a context, a question, and four answer options. In order to determine the most suitable answer option, it is crucial to acquire a comprehensive understanding of the logical information embedded within the sentences $\{s_1, s_2, ..., s_8\}$ of context and effectively integrate them through a series of logical operations.

Given the remarkable performance of large-scale pre-trained language models (Devlin et al.; Liu et al., 2019b; Yang et al., 2019; Lan et al.; He et al.) in text comprehension, previous research has progressed towards the development of methods that delve into contextual contexts to extract and analyze logical information. Notably, methods such as DAGN (Huang et al.), FocalReasoner (Ouyang et al., 2021) and HGN (Chen et al., a) adopt a strategy of dividing the context and options into distinct units. These units are then integrated through

* Corresponding author

Findings of the Association for Computational Linguistics: NAACL 2024, pages 1945–1955
June 16-21, 2024

the construction of specialized graphs, facilitating the process of reasoning. In contrast, LReasoner (Wang et al.) employs a context extension framework based on symbolic logic, supplemented by the utilization of data augmentation methods to predict the answer.

However, these methods rely on embedding text into low-dimensional vectors and performing reasoning in Euclidean space, which introduces certain limitations. 1) The logical information represented in text across different sentences often exhibits uncertainty, which is challenging to capture through a single vector representation. This uncertainty refers to the core logical content that sentences emphasize when interacting with each other is different. As illustrated in Figure 1, the blue clauses denote critical components where logical information intersects between s_1, s_4, and s_7, similar to the green clauses between s_1 and s_5, as well as the orange clauses between s_1 and s_8. 2) Handling logical operations (such as $\cup, \cap, and \neg$) in an interpretable manner in Euclidean space remains an unresolved issue. Previous works perform vaguely through simple arithmetic operations on vectors to integrate logical information. Indeed, this information should be represented as a logical union operation on multiple sentences.

Motivated by the probabilistic embedding in knowledge graph (Ren and Leskovec, 2020), we propose Beta-LR, a novel probabilistic embedding method designed for logical reasoning based on Beta distribution. Our Beta-LR learns the logical information within the context and models logical operators using beta distributions with bounded support. To ensure the preservation of logical integrity, we first employ a syntactic division of the context based on punctuation, avoiding any disruption caused by more fine-grained splitting. We then embed the sentences from the divided context, along with the question and options, as beta distributions defined on the [0, 1] interval, effectively eliminating logical uncertainty. Additionally, we define logical intersection and negation operators to facilitate logical union operations based on De Morgan's laws, enabling the interpretable integration of logical information. In this process, our beta-LR demonstrates unique advantages in modeling logical negation operation compared with other probabilistic embedding methods. Finally, the integrated logical information is utilized to predict the best answer.

We conduct experiments on two logical reasoning datasets: ReClor (Yu et al.) and LogiQA (Liu et al.). We utilize RoBERTa(Liu et al., 2019b) and DeBERTa(He et al.) as our backbone pre-trained model for evaluation. The experimental results showcase the competitiveness of our approach and its efficacy in capturing logical information within textual data for reasoning purposes. Our contributions are three folds:

1) We propose a novel approach that leverages beta distributions for capturing logical information. And we define interpretable logical operators that ensure reliable and meaningful integration of logical information, enabling more accurate reasoning.

2) Through experiments conducted on two datasets, we demonstrate the competitiveness of our approach. Additionally, our ablation studies highlight the significance of interpretable logical operators and relative modules in effectively integrating contextual information.

3) Our case study experiment demonstrates that embedding text into beta distributions can effectively capture and mitigate the uncertainty of textual information. This highlights the effectiveness of our Beta-LR.

2 Related Work

2.1 Logical Reasoning of Text

Previous researchers have dedicated significant efforts to logical reasoning tasks. In terms of model design, DAGN (Huang et al.) introduced a network that extracts discourse units from text and performs chain reasoning using graph networks. FocalReasoner (Ouyang et al., 2021) constructs facts by extracting core components from sentences and builds a hypergraph to facilitate interaction at the sentence and entity levels. HGN (Chen et al., a) employs a holistic graph network to process context at the discourse and word levels, enabling finer-grained relationship extraction for logical reasoning. In terms of data augmentation, LReasoner (Wang et al.) proposes a template-based method to convert logical expressions into text, expanding the dataset through extended text. MERIt (Jiao et al.) adopts a Meta path guided contrastive learning approach for self-supervised pre-training on rich unlabeled text data, which benefits downstream reasoning tasks. While these models and methods have achieved notable results, they have also encountered limitations due to the limited ability of traditional pre-trained models. Building on this, our proposed method focuses on enhancing the embedding representation's capability to capture logical information, starting from the level of data embedding representation.

2.2 Probabilistic Embedding

Several notable approaches in the field of probabilistic embedding have been proposed for uncertain knowledge graph reasoning (Ren and Leskovec, 2020; Yang et al., 2022; Wang et al., 2022). For example, Query2Box (Ren et al.) introduces a probabilistic framework that models uncertainty in knowledge graph embeddings by representing entities and relations as boxes in a high-dimensional space. ConE (Zhang et al., 2021) proposes a method representing entities and relations as cones in a high-dimensional space to capture the uncertainty in the existence of certain relationships between entities. BEUrRE (Chen et al., b) extends the Query2box framework to handle uncertain knowledge graph reasoning. These works collectively contribute to the advancement of probabilistic embedding techniques.

3 Preliminary

3.1 Problem

We address the problem of logical reasoning in the Machine Reading Comprehension (MRC) task (Zhang et al., 2019; Liu et al., 2019a). Our objective is to develop a model that can effectively extract and reason with logical information from a given dataset. The dataset consists of a context C, which provides a background or a passage of text, a question Q that needs to be answered using the information in the context, and four corresponding options O_1, O_2, O_3, O_4. The challenge lies in identifying the logical structure and relationships within the context and leveraging this information to select the most appropriate option O_a that correctly answers the question.

3.2 Beta Distribution

The beta distribution is a continuous probability distribution defined on the interval [0, 1]. The distribution is characterized by two shape parameters, commonly denoted as Beta(α, β) $(\alpha > 0, \beta > 0)$. Our methodology extensively leverages its probability density function (PDF) expressed as equation 1:

$$p(x) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha, \beta)} \tag{1}$$

where $x \in [0, 1]$ and $B(\cdot)$ is the beta function. The uncertainty of a Beta distribution can be measured by its entropy: $H = \ln B(\alpha, \beta) - (\alpha - 1)[\psi(\alpha) - \psi(\alpha+\beta)] - (\beta-1)[\psi(\beta) - \psi(\alpha+\beta)]$, where $\psi(\cdot)$ represents the digamma function.

We leverage two crucial properties of the beta distribution in our approach. 1) When the beta distribution is used as the prior distribution for a parameter in Bayesian inference, the posterior distribution obtained after incorporating observed data also follows a beta distribution. 2) The beta distribution allows for the overlay or stacking of multiple beta distributions, enabling the creation of composite distributions.

4 Methodology

To enhance our ability to capture and utilize logical information, we introduce a novel probabilistic embedding method, Beta-LR. The architecture of Beta-LR is detailed in Figure 2. The process begins with segmenting the context into individual sentence units and encoding them using beta distributions, as discussed in section 4.1. We then proceed to develop and apply logical operators, which allow us to integrate and interpret the logical information across these sentences, creating a cohesive logical representation of the context in section 4.2. In the final step, outlined in section 4.3, we integrate this comprehensive context representation with the given question and options, enabling accurate answer prediction.

4.1 Encoder

To effectively extract logical information at a granular level and ensure its integrity, we employ syntactic techniques to divide the context into fundamental sentence units. The context, represented as C, is split into a collection of sentences $\{s_1, s_2, \ldots, s_N\}$ using punctuation as the basis for segmentation. For each option O_j, we utilize RoBERTa, a pre-trained language model, to embed token sequences. These sequences are formed by concatenating $\{\texttt{<s>}s_1\texttt{||}s_2\texttt{||}\ldots\texttt{||}s_N\texttt{</s>}Q\texttt{</s>}O_j\texttt{</s>}\}$, where $\{\texttt{||}, \texttt{</s>}\}$ act as separators in RoBERTa. Given the token sequence $\{t_1^i, t_2^i, \ldots, t_K^i\}$ with length K for s_i, the output embedding $\{v_{t_1}^i, v_{t_2}^i, \ldots, v_{t_K}^i\}$ is averaged to form the vector representations of s_i in Euclidean space:

$$E_{s_i} = \frac{1}{|K|}\sum_{j=1}^{|K|} v_{t_j}^i \tag{2}$$

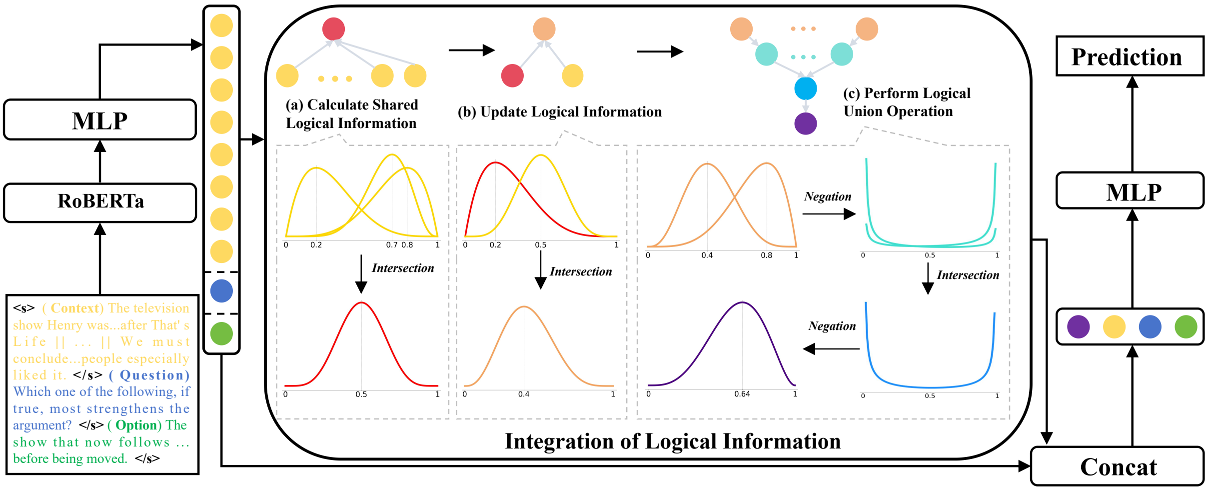

Figure 2: The overall architecture of our proposed model, Beta-LR. For logical reasoning, each sentence, along with the question and options, is encoded into beta distributions. Subsequently, these distributions undergo three steps to effectively integrate the logical information for answer prediction. (a) Calculate shared logical information. (b) Update logical information. (c) Perform logical union operation. We depict beta distribution on each dimension by different points to demonstrate the integration of logical information. The representation of beta distribution for different points are shown in Figure 3. And the gray arrows indicate the process of logical operations.

Points	Representation of Beta Distribution
	Logical information of sentence s_i
	Logical information of question Q
	Logical information of option O_j
	Shared logical information obtained through logical intersection operation on several
	Updated logical information obtained through logical intersection operation on and
	Opposite logical information of each
	Shared logical information obtained through logical intersection operation on several
	Integrated logical information of context

Figure 3: The illustration of representation of beta distribution for different points.

Due to the limitations of traditional embeddings in resolving the ambiguity of logical information in context and performing interpretable logical operations, we advocate for the use of bounded supported probability embeddings. Specifically, we map text representations into a beta distribution along each dimension. We aim to generate a novel embedding $B_{s_i} = [(\alpha_1, \beta_1), (\alpha_2, \beta_2), \ldots, (\alpha_d, \beta_d)] \in \mathbb{R}^{2d}$, where the parameters correspond to the d-dimensional beta distribution. The embedding B_{s_i} is derived on the vector representation following equation 3:

$$B_{s_i} = 1 + \sigma(f(E_{s_i}; \theta)), \tag{3}$$

where $f(\cdot)$ denotes a Multi-Layer Perceptron, and θ represents its associated parameters. Notably, we use the rectified linear unit (ReLU) activation function σ on the beta distribution parameters and add a constant of 1 to ensure the parameters remain stable within a suitable range. This technique effectively prevents the parameters of the beta distribution from becoming zero after passing through the ReLU and helps control the initial parameter values of B_{s_i} between 1 and 2 ensuring a more reliable convergence during the training process. By embedding text into beta distributions, we effectively capture and encapsulate logical information in a robust and nuanced manner, significantly enhancing the interpretability and comprehension of inherent logical relationships in the data.

To facilitate the utilization of B_{s_i} for logical operations, we partition it into two components $B^{\alpha}_{s_i} = [\alpha_1, \alpha_2, ..., \alpha_d]$ and $B^{\beta}_{s_i} = [\beta_1, \beta_2, ..., \beta_d]$. These components represent the parameter vectors α and β, respectively. Thus, B_{s_i} can be expressed as $B_{s_i} = [B^{\alpha}_{s_i} || B^{\beta}_{s_i}] \in R^{2d}$. Similarly, employing the same approach, we can obtain embeddings $B_Q = [B^{\alpha}_{Q} || B^{\beta}_{Q}] \in R^{2d}$ and $B_{O_i} = [B^{\alpha}_{O_j} || B^{\beta}_{O_j}] \in R^{2d}$ to represent Q and O_j, respectively.

4.2 Integration of Logical Information

In the context C, each s_i inherently contains crucial logical information, which is a significantly important aspect. The integration of this logical information is vital for enhancing logical clarity and improving answer prediction accuracy.

Calculate Shared Logical Information To emphasize important details, often referred to as shared logical information (SLI), and to enhance the logical expression of s_i, we propose the modeling of a logical intersection operator, denoted as $\mathcal{I}$. This operator is designed to compute a new embedding that encapsulates SLI. As depicted in Figure 4, the objective of $\mathcal{I}$ is to calculate a new embedding B_{inter}, representing the intersection of the distributions in the given set of n input embeddings $B_{s_1}, B_{s_2}, \ldots, B_{s_N}$:

$$B_{inter} = \mathcal{I}(B_{s_1}, B_{s_2}, \ldots, B_{s_N}) \tag{4}$$

We model the intersection operator $\mathcal{I}$ by taking the weighted product of the PDFs of the input embeddings based on the additivity of beta distributions. This approach intuitively aligns with the assumption that regions exhibiting high density in the new distribution should also exhibit high density in all input distributions. It has been shown that when $B_{s_i} = [(\alpha_1, \beta_1), (\alpha_2, \beta_2), \ldots, (\alpha_d, \beta_d)] \in \mathbb{R}^{2d}$ represents d-dimensional Beta distributions, the weighted product of PDFs can be viewed as a linear interpolation of the parameters of the inputs (Ren and Leskovec, 2020). Therefore, the parameters of B_{inter} can be described as $[(\sum w_i \odot \alpha, \sum w_i \odot \beta)] \in \mathbb{R}^{2d}$, where $\alpha \in \mathbb{R}^{N \times d}$ and $\beta \in \mathbb{R}^{N \times d}$. Here, $w_i \in \mathbb{R}^d$ and $\odot$ denotes the dimension-wise product. The weights w_i are calculated using an MLP-based attention mechanism as defined in Equation 5:

$$w_i = \frac{\exp(\text{MLP}_{\text{Att}}([B_{s_i}^{\alpha} \,||\, B_{s_i}^{\beta}]))}{\sum_{j=1}^{N} \exp(\text{MLP}_{\text{Att}}([B_{s_j}^{\alpha} \,||\, B_{s_j}^{\beta}]))}, \tag{5}$$

where $\text{MLP}_{\text{Att}}(\cdot) : \mathbb{R}^{2d} \to \mathbb{R}^d$ is a Multi-Layer Perceptron. Through this mechanism, the operator $\mathcal{I}$ effectively captures the strong correlation and consistency between SLI and the logical information represented by each Beta distribution. Employing $\mathcal{I}$, we compute the embedding B_{inter} that encapsulates SLI, as illustrated in Figure 2 (a).

Update Logical Information In the subsequent step, as shown in Figure 2 (b), B_{inter} is employed to update B_{s_i} based on the conjugate prior property of the beta distribution. We interpret the initial embedding B_{s_i} and B_{inter} as representations of the prior distribution and the likelihood function, respectively. Then, we calculate the posterior distribution, denoted as B'_{s_i}, by taking the weighted product of the probability density functions of B_{inter} and B_{s_i}, as shown in Equation 6. It is important to note that while the calculation of B'_{s_i} is conceptually different from that of B_{inter}, they follow a similar computational procedure.

$$B'_{s_i} = \mathcal{I}(B_{s_i}, B_{inter}) \tag{6}$$

Perform Logical Union Operation Lastly, to model the logical union of B'_{s_i} and obtain a new embedding that represents the integrated logical information of the context, we face the challenge of defining an interpretable logical union operator. To address this, we transform union operations into a combination of intersection and negation operations, leveraging De Morgan's laws (Saha, 2022). This approach allows us to effectively model the logical union in a computationally feasible manner.

Considering logical negation operations, as depicted in Figure 5, we propose a probabilistic negation operator $\mathcal{N}$ that operates specifically on the embedding representation B'_{s_i} of s_i, generating an alternative embedding representation $\widehat{B'_{s_i}}$ that encapsulates the opposite logical information. The formulation of this operator is expressed in Equation 7:

$$\widehat{B'_{s_i}} = \mathcal{N}(B'_{s_i}) \tag{7}$$

Leveraging the distinctive properties of the beta distribution's probability density function, we formalize the logical negation operator by taking the reciprocal of the beta distribution's two parameters: $\mathcal{N}[(\alpha, \beta)] = \left[\left(\frac{1}{\alpha}, \frac{1}{\beta}\right)\right]$. This method ensures that regions with high probability density in the original distribution correspond to low-density regions in the negated output, and vice versa, providing strong interpretability for the logical negation operator.

As illustrated in Figure 2 (c), given N embeddings $\{B'_{s_1}, B'_{s_2}, \ldots, B'_{s_N}\}$, we compute a new embedding B_{union} to represent the integrated logical information of the context. This calculation employs the operators $\mathcal{I}$ and $\mathcal{N}$ as per Equation 8:

$$B_{union} = \mathcal{N}(\mathcal{I}(\mathcal{N}(B'_{s_1}), \mathcal{N}(B'_{s_2}), \ldots, \mathcal{N}(B'_{s_N}))) \tag{8}$$

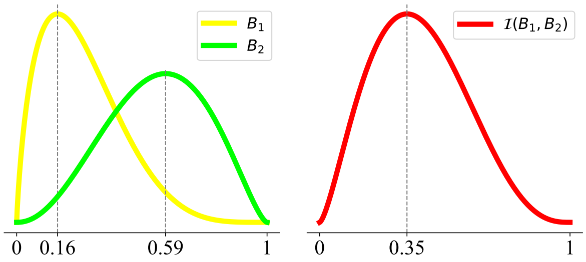

Figure 4: Illustration of intersection operator $\mathcal{I}$. $\mathcal{I}$ is targeted to compute a new embedding that encapsulates SLI by taking the weighted product of the PDFs.

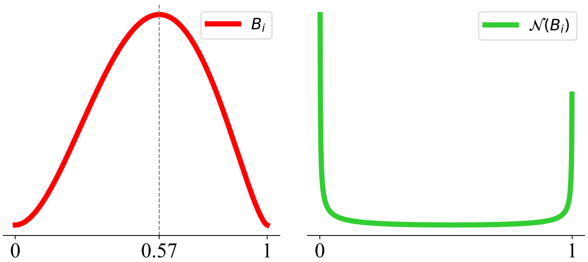

Figure 5: Illustration of the negation operator $\mathcal{N}$. $\mathcal{N}$ generates a new embedding encapsulating the opposite logical information by taking the reciprocal of its parameters.

4.3 Answer Prediction

After integrating the logical information from the context, we obtain the embedding B_{union}, which is then inputted into the answer prediction module. Recognizing the significance of the initial global feature, we calculate the average of the initial embeddings of all tokens within the context, resulting in the embedding $B_C \in \mathbb{R}^{2d}$. Furthermore, we acquire the embeddings B_Q and B_{O_j} as discussed in Section 2.1. To construct the final representation, we concatenate the embeddings $B_{union} \in \mathbb{R}^{2d}$, $B_C \in \mathbb{R}^{2d}$, $B_{O_j} \in \mathbb{R}^{2d}$, and $B_Q \in \mathbb{R}^{2d}$ to form $B_{cat} \in \mathbb{R}^{8d}$. Subsequently, B_{cat} is passed through a Multi-Layer Perceptron to acquire the predicted probability p_j for each option j. The option with the highest probability is deemed the correct option O_a. To train our model in an end-to-end manner, we employ cross-entropy loss.

5 Experiment

We evaluate the performance of our Beta-LR on two datasets, ReClor (Yu et al.) and LogiQA (Liu et al.). Furthermore, we conduct an ablation study to examine the effectiveness of logical operators and crucial modules. During the training process, the AdamW (Loshchilov and Hutter, 2017) with $\beta 1 = 0.9$ and $\beta 2 = 0.99$ is taken as the optimizer and batch size is set to 8. The learning rate is 1e-6 for ReClor and 5e-6 for LogiQA. The model is trained for 20 epochs and 10 epochs on ReClor and LogiQA respectively, to obtain the optimal results.

5.1 Dataset and Baselines

The ReClor dataset comprises a total of 6,138 examples, with 4,638 examples dedicated to training, 500 examples for validation, and 1,000 examples for testing. The test examples are further divided into two categories: EASY examples and HARD examples. We assess the performance of our model on the test set, as well as on the EASY and HARD subsets separately. The LogiQA dataset contains 8678 samples, which have been randomly partitioned into training, validation, and testing sets, consisting of 7,376, 651, and 651 samples respectively.

To enable effective comparison with previous studies, we utilize RoBERTa-large and DeBERTa-xlarge as our backbone model with employing accuracy as the evaluation metric. In addition, our method mainly aims to identify the logical content and capture logical information of context. In order to avoid the impact of additional data processing techniques, we have selected the methods without data augmentation as baselines, which including pre-trained language models (Liu et al., 2019b), DAGN (Huang et al.), FocalReasoner (Ouyang et al., 2021), HGN (Chen et al., a),LReasoner (Wang et al.) and Logiformer(Xu et al.). Notably, we compare with Logiformer only on syntax graph branch because logical graph branch is not designed in our model.

5.2 Results

Table 1 and Table 2 report the best experimental results on RoBERTa backbone and DeBERTa backbone respectively. We observe varying degrees of improvement compared to baseline models. On the ReClor dataset, Beta-LR achieves 6.7% increase and 7.2% increase on valid sets and test sets. Simultaneously, our method demonstrates a superior capability to solve challenging problems, as evidenced by a remarkable 12.0% improvement on HARD subsets compared to a modest 3.8% improvement on EASY subsets. On the LogiQA dataset, Beta-LR also shows remarkable improvements with achieving 13.1% increase in validation accuracy and 15.3% increase in test accuracy. Additionally, our method achieves consistent improvement over DeBERTa backbone,with 3.2% in vali-

Model	ReClor				LogiQA	
	Valid	Test	Test-E	Test-H	Valid	Test
Random	25.0	25.0	25.0	25.0	25.0	25.0
RoBERTa	62.6	55.6	75.5	40.0	35.0	35.3
DAGN	65.2	58.2	76.1	44.1	35.5	38.7
FocalReasoner	**66.8**	58.9	77.1	44.6	**41.0**	40.3
HGN	66.4	58.7	77.7	43.8	40.1	39.9
LReasoner	65.2	58.3	**78.6**	42.3	-	-
Logiformer	63.6	**59.9**	75.0	**48.0**	38.3	37.6
Beta-LR(RoBERTa)	**66.8**($\uparrow$ 6.7%)	59.6($\uparrow$ 7.2%)	78.4($\uparrow$ 3.8%)	44.8($\uparrow$ 12.0%)	39.6($\uparrow$ 13.1%)	**40.7**($\uparrow$ 15.3%)

Table 1: Experimental results (Accuracy: %) of Beta-LR compared with baseline models on RoBERTa backbone. Test-E and Test-H are the EASY subset and HARD subset on ReClor, respectively. The results of each baseline model align with the findings reported in their respective published papers. The content within the parentheses $(\cdot)$ represents the improvement compared to RoBERTa.

Model	ReClor	
	Valid	Test
DoBERTa	74.4	68.9
LReasoner	74.6	71.8
HGN	76.0	72.3
Beta-LR(DeBERTa)	**76.8**($\uparrow$ 3.2%)	**72.9**($\uparrow$ 5.8%)

Table 2: Experimental results (Accuracy: %) of Beta-LR compared with baseline models on DeBERTa backbone.

Model	Valid	Test
Beta-LR(RoBERTa)	66.8	59.6
-w/o embedding in beta distribution	64.0	56.2
-w/o update of logical information	65.8	57.9
-w/o weighted product	63.2	57.1

Table 3: Ablation results (Accuracy: %) on ReClor.

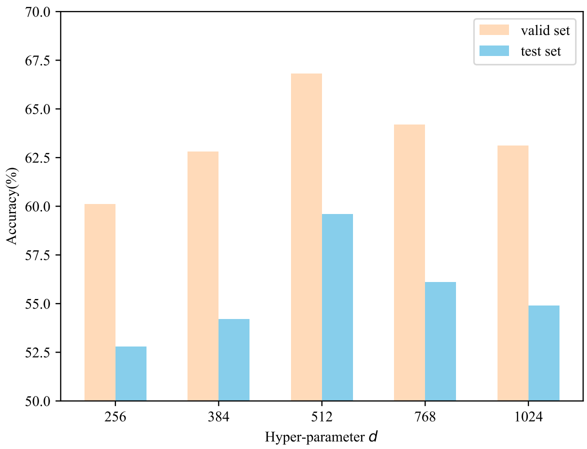

Figure 6: The experimental results under different hyper-parameter d.

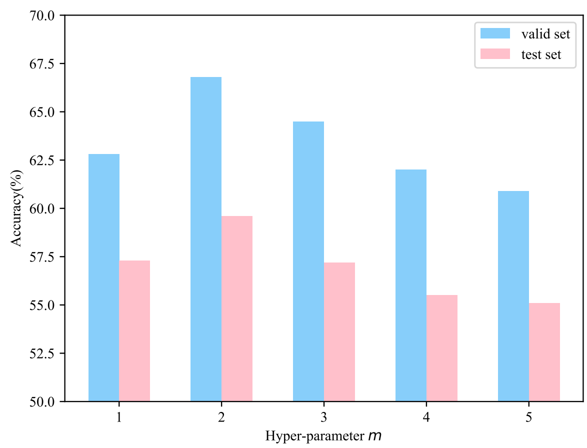

Figure 7: The experimental results under different hyper-parameter m.

dation accuracy and 5.8% increase in test accuracy on Reclor.

5.3 Ablation Study

We conduct a series of ablation studies on three aspects to verify the importance of logical operators and relative modules. The results are shown in Table 3.

Embedding in Beta Distribution We replaced the embedding of text into beta distributions with vectors in Euclidean space to analyze the ability of Beta-LR to capture logical information for reasoning. For logical intersection and union operation, we employed weighted attention and arithmetic averaging on vectors as replacements. And negation operation is not required. The results demonstrated a significant decline in performance due to these replacements. The accuracy dropped to 64.0% and 56.2% on the valid and test sets. This indicates that embedding text into vectors alone cannot effectively capture logical information, and the absence of dedicated logical operators hinders the integration of logical information.

Update of Logical Information We remove the step that calculates B_{inter} and update the logical information of each sentence during the process of integrating logical information. Instead, we utilized the initial logical information to represent the sentences directly, completing the union operation to generate B_{union}. The accuracy witnessed a decrease of 1.0% and 1.7% on two sets. This outcome serves as evidence that calculating shared logical information is indeed necessary and updating the logical information to obtain a more refined representation proves to be beneficial for performing logical reasoning.

Weighted Product We replace the weighted product of PDFs in the intersection operation with a simple average. As a result, the performance drops to 63.2% and 57.1% on valid and test sets respectively. It proves that by employing the weighted product, the model can better focus on key information, enabling the generation of more comprehensive and accurate text logical expressions.

5.4 Parameter Analysis

In Beta-LR, we employ the embedding of text into beta distributions, where the embedding size d plays a crucial role as a significant hyper-parameter that impacts the overall outcome. To investigate the influence of parameter d, we conducted an in-depth analysis, and the results are presented in Figure 6. The findings clearly indicate that our Beta-LR with $d = 512$ achieves the best performance for both the valid and test sets. A lower embedding size fails to adequately express logical information, while a higher embedding size introduces more redundant information, thereby compromising the reasoning ability, which provides a plausible explanation for the observed outcomes.

During the intersection operation process, the MLP_{Att} plays a crucial role. We study the impact of the layer number, denoted as m, by considering five sets of hyper-parameters for m. The accuracy results are presented in Figure 7. It is evident that the experimental performance is optimal when the layer size is set to 2. Specifically, as m increases, the accuracy of the experimental results decreases on the validation and test sets. This observation aligns with the explanation that an excessive number of layers carries a certain risk of overfitting and integrating unnecessary logical information.

Model	Number of parameters
Roberta-large	355M
DAGN	400M
FocalReasoner	414M
Beta-LR(RoBERTa)	360M

Table 4: The comparison results with the number of parameters.

5.5 Case Study

We analyze the data example introduced in Figure 1 to verify the ability of our Beta-LR model in capturing the uncertainty of logical information within the context. It is evident that the level of uncertainty in the logical information conveyed by a sentence tends to increase with the number of words it contains. For each sentence s_i, we record the word count as W_i. Subsequently, we calculate the final logical representations B'_{s_i} for s_i, which are represented as multidimensional beta distributions. The average entropy of these beta distributions, denoted as E_i, is calculated to quantify the uncertainty. A Spearman's correlation test is conducted between these two sets of data. The test results in a correlation coefficient of 0.762 with a significance level of $p < 0.05$, as presented in Figure 8. This statistically significant result underscores the strong relationship between the two data sets, thereby demonstrating the effectiveness of our proposed method in capturing the uncertainty of logical information within sentences.

5.6 Size Analysis

We analyze the parameter count in our model to confirm its scalability advantage. The comparison results, displayed in Table4, indicate the number of parameters in relation to the baseline model. Notably, our model showcases a mere addition of 5M parameters at the RoBERTa scale, which is substantially smaller than that of other baseline models. This observation demonstrates the unification of complexity and accuracy in our model.

5.7 Error analysis

We conducted error analysis on the ReClor dataset. This dataset integrates various logical reasoning skills and can be divided into 17 types. The detailed results of different types of logical reasoning are shown in Table5. Compared to the baseline, our

Context	(s_1) The television show Henry was not widely watched until it was scheduled for Tuesday evenings immediately after That' s Life, (s_2) the most popular show on television. (s_3) During the year after the move, (s_4) Henry was consistently one of the ten most-watched shows on television. (s_5) Since Henry' s recent move to Wednesday evenings, (s_6) however, (s_7) it has been watched by far fewer people. (s_8) We must conclude that Henry was widely watched before the move to Wednesday evenings because it followed That' s Life and not because people especially liked it.							
Sentence	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8
W_i	19	6	5	11	7	1	8	25
E_i	0.9623	0.8968	0.8674	1.0296	1.0696	0.8968	1.1099	1.1352
Spearman's correlation	0.762							
p	0.028(<0.05)							

Figure 8: The results of Spearman correlation coefficient and p-value of two sets of data W_i and E_i.

Reasoning Type	RoBERTa-large	Beta-LR
Centered Necessary Assumptions	71.0	66.7(↓)
Sufficient Assumptions	46.7	63.3(↑)
Strengthen	61.7	59.6(↓)
Weaken	47.8	54.9(↑)
Evaluation	69.2	76.9(↑)
Implication	39.1	43.5(↑)
Conclusion/Main Point	63.9	58.3(↓)
Most Strongly Supported	42.9	53.6(↑)
Explain or Resolve	58.3	63.1(↑)
Principle	50.8	67.7(↑)
Dispute	50.0	63.3(↑)
Technique	52.8	66.7(↑)
Role	56.2	56.3(↑)
Identify a Flaw	61.5	64.1(↑)
Match Flaws	45.2	29.0(↓)
Match the Structure	56.7	60.0(↑)
Others	52.1	57.5(↑)

Table 5: The results on different logical reasoning types. ↓, ↑ respectively mean that Beta-LR is better and worse than baseline model.

model has made significant improvements in most types of logical reasoning, but performs poorly in the following problem types: Centered Necessity Assumptions, strength, Conclusion/Main Point and Match Flaws. These problems revolve identifying contextually relevant information that strengthens reasoning or arguments, which exist noticeable semantic gap between context and arguments. How to improve the ability to comprehend and align deep semantic information will be a key focus area in our future work.

6 Conclusion

In this paper, we introduce Beta-LR, a probabilistic embedding method for interpretable logical reasoning. Our work is the first to address logical reasoning at the level of data embedding representation. Through experiments conducted on two datasets, we demonstrate that Beta-LR achieves competitive performances in logical reasoning tasks. The results validate the effectiveness of our method in capturing and utilizing logical information for reasoning purposes. By addressing the challenges of logical uncertainty and the fusion of logical information, Beta-LR provides a valuable solution for enhancing logical reasoning capabilities.

Limitations There could be two limitations to our approach. Firstly, Our Beta-LR only calculates the shared logical information among all sentences, but overlooks the shared components between any two sentences. Additionally, representing large amounts of textual logical information using bounded beta distributions still presents challenges. In future research, we will explore more comprehensive probabilistic embedding methods to effectively learn logical information from text.

Acknowledgement This work was supported by the National Natural Science Foundation of China (grants No.62176046), and the Sichuan Science and Technology Program (No.2022ZHCG0008 and No.2023JDRC0086).

References

Jialin Chen, Zhuosheng Zhang, and Hai Zhao. a. Modeling hierarchical reasoning chains by linking discourse units and key phrases for reading comprehension. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 1467–1479.

Xuelu Chen, Michael Boratko, Muhao Chen, Shib Sankar Dasgupta, Xiang Lorraine Li, and Andrew McCallum. b. Probabilistic box embeddings for uncertain knowledge graph reasoning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computa-*

tional Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021, pages 882–893.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. The argument reasoning comprehension task: Identification and reconstruction of implicit warrants. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1930–1940.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: decoding-enhanced bert with disentangled attention. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

Yinya Huang, Meng Fang, Yu Cao, Liwei Wang, and Xiaodan Liang. DAGN: discourse-aware graph network for logical reasoning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 5848–5855.

Fangkai Jiao, Yangyang Guo, Xuemeng Song, and Liqiang Nie. Merit: Meta-path guided contrastive learning for logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 3496–3509.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Hanmeng Liu, Jian Liu, Leyang Cui, Zhiyang Teng, Nan Duan, Ming Zhou, and Yue Zhang. 2023. Logiqa 2.0 - an improved dataset for logical reasoning in natural language understanding. *IEEE ACM Trans. Audio Speech Lang. Process.*, 31:2947–2962.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3622–3628.

Shanshan Liu, Xin Zhang, Sheng Zhang, Hui Wang, and Weiming Zhang. 2019a. Neural machine reading comprehension: Methods and trends. *CoRR*, abs/1907.01118.

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

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

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

Siru Ouyang, Zhuosheng Zhang, and Hai Zhao. 2021. Fact-driven logical reasoning. *CoRR*, abs/2105.10334.

Hongyu Ren, Weihua Hu, and Jure Leskovec. Query2box: Reasoning over knowledge graphs in vector space using box embeddings. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Hongyu Ren and Jure Leskovec. 2020. Beta embeddings for multi-hop logical reasoning in knowledge graphs. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Ranjini Saha. 2022. Study of application of de-morgan's law in modern fields. *Journal of Applied Science and Education (JASE)*, 2:1–8.

Kai Wang, Chunhong Zhang, Jibin Yu, and Qi Sun. 2022. Signal embeddings for complex logical reasoning in knowledge graphs.

Siyuan Wang, Wanjun Zhong, Duyu Tang, Zhongyu Wei, Zhihao Fan, Daxin Jiang, Ming Zhou, and Nan Duan. Logic-driven context extension and data augmentation for logical reasoning of text. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 1619–1629.

Fangzhi Xu, Jun Liu, Qika Lin, Yudai Pan, and Lingling Zhang. Logiformer: A two-branch graph transformer network for interpretable logical reasoning. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, pages 1055–1065.

D. Yang, Peijun Qing, Y. Li, H. Lu, and Xiaodong Lin. 2022. Gammae: Gamma embeddings for logical queries on knowledge graphs. In *Conference on Empirical Methods in Natural Language Processing*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5754–5764.

Weihao Yu, Zihang Jiang, Yanfei Dong, and Jiashi Feng. Reclor: A reading comprehension dataset requiring logical reasoning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Xin Zhang, An Yang, Sujian Li, and Yizhong Wang. 2019. Machine reading comprehension: a literature review. *CoRR*, abs/1907.01686.

Zhanqiu Zhang, Jie Wang, Jiajun Chen, Shuiwang Ji, and Feng Wu. 2021. Cone: Cone embeddings for multi-hop reasoning over knowledge graphs. In *Neural Information Processing Systems*.