Empowering Tree-structured Entailment Reasoning: Rhetorical Perception and LLM-driven Interpretability

Longyin Zhang, Bowei Zou, Ai Ti Aw
Institute for Infocomm Research, A*STAR, Singapore
{zhang_longyin,zou_bowei,aaiti}@i2r.a-star.edu.sg

Abstract

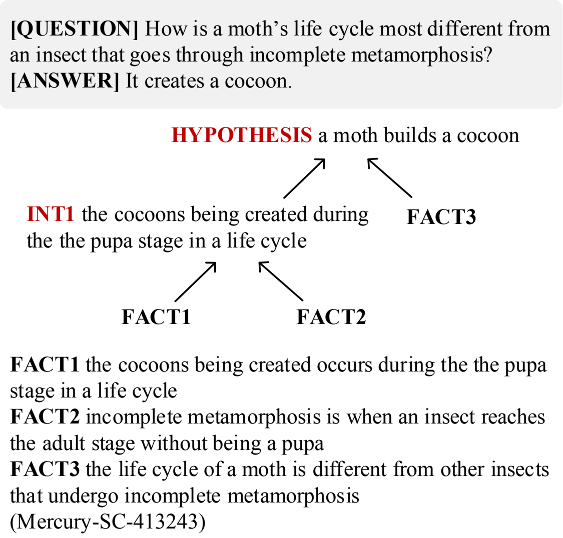
The study delves into the construction of entailment trees for science question answering (SQA), employing a novel framework termed **T**ree-structured **E**ntailment **R**easoning (TER). Current research on entailment tree construction presents significant challenges, primarily due to the ambiguities and similarities among candidate science facts, which considerably complicate the fact retrieval process. Moreover, the existing models exhibit limitations in effectively modeling the sequence of reasoning states, understanding the intricate relations between neighboring entailment tree nodes, and generating intermediate conclusions. To this end, we explore enhancing the TER performance from three aspects: First, improving retrieval capabilities by modeling and referring to the chained reasoning states; Second, enhancing TER by infusing knowledge that bridges the gap between reasoning types and rhetorical relations. Third, exploring a task-specific large language model tuning scheme to mitigate deficiencies in intermediate conclusion generation. Experiments on the English EntailmentBank demonstrate the effectiveness of the proposed methods in augmenting the quality of tree-structured entailment reasoning to a certain extent.

Keywords: Tree-structured Entailment Reasoning, Science QA, Entailment Tree Construction

1. Introduction

Traditional science question-answering (SQA) approaches often treat the QA process as a black box, where a model takes a question as input and leverages its reasoning ability and linguistic knowledge to produce an answer directly. This approach, while effective, obfuscates the inner workings of the model's decision-making process, making it challenging to understand how the model reaches the answers. To this limitation, recent research has emphasized the importance of generating explanations, which can illuminate the intricate and implicit relations between questions and answers. This has spurred significant studies in constructing support facts (Mihaylov et al., 2018; Khot et al., 2020), entailment trees (Clark et al., 2018; Dalvi et al., 2021), explanation graphs (Jansen et al., 2018), and reasoning chains (Lu et al., 2022; Wei et al., 2022) to simulate the chain-of-thought inference process like humans. This paper focuses on the realm of **T**ree-structured **E**ntailment **R**easoning (TER), which facilitates a deeper understanding of the mechanisms of SQA (Dalvi et al., 2021).

Previous research on entailment tree construction predominantly falls into two distinct approaches: one-step entailment tree analysis (Bentivogli et al., 2011; Bowman et al., 2015) and multi-step entailment tree construction (Dalvi et al., 2021). This paper places a particular emphasis on the latter approach. The multi-step entailment tree construction task involves the provision of a hypothesis that explains the question-answering (QA) pair. As illustrated in Figure 1, given the hypothesis and the three supporting facts retrieved from the science fact pool, the primary objective of TER is to parse these facts into a hierarchical entailment tree which serves as an explanatory framework for the given QA pair.



[QUESTION] How is a moth's life cycle most different from an insect that goes through incomplete metamorphosis?
[ANSWER] It creates a cocoon.

HYPOTHESIS a moth builds a cocoon
INT1 the cocoons being created during the the pupa stage in a life cycle
FACT3
FACT1
FACT2

FACT1 the cocoons being created occurs during the the pupa stage in a life cycle
FACT2 incomplete metamorphosis is when an insect reaches the adult stage without being a pupa
FACT3 the life cycle of a moth is different from other insects that undergo incomplete metamorphosis
(Mercury-SC-413243)

Figure 1: Example tree from EntailmentBank.

Recently, some excellent research has been conducted on multi-step entailment tree building. Initially, Tafjord et al. (2020) and Dalvi et al. (2021) cast the TER task as a sequence-to-sequence generation task; they take all the facts as input to the pre-trained T5 transformer (Raffel et al., 2020) to generate the entailment tree all at once. Since the probability distribution inside the entailment steps is invisible to the sequential model, the sequence-

LREC-COLING 2024, pages 5783–5793
20-25 May, 2024. © 2024 ELRA Language Resource Association: CC BY-NC 4.0

to-sequence approaches may generate unreliable intermediate steps. In order to mitigate this issue, more and more subsequent research proposes to transform the TER task into a multi-step tree construction process, to generate explanations step by step (Ribeiro et al., 2022; Bostrom et al., 2022; Liu et al., 2022) from the bottom up. Besides, (Hong et al., 2022) introduces the first module-based framework that generates entailment trees bidirectionally through the top-down abductive and bottom-up deductive reasoning steps.

In general, the TER task involves two stages, with the first stage to retrieve relevant facts as premises and the second to build the entailment tree with the premises acting as leaf nodes. Although considerable efforts have been made to enhance the retrieval ability of recent multi-step reasoning systems (Ribeiro et al., 2022), there remains room for improvement in performance. Moreover, previous work (Hong et al., 2022) has underscored the effectiveness of reasoning types in building entailment trees, prompting further investigation into how discourse information functions on entailment reasoning. In particular, recent years have witnessed the extraordinary prowess of Large Language Models (LLMs) in natural language generation, yet TER based on instruction-tuned LLMs remains largely unexplored.

With these inspirations, in this paper, based on the multi-module TER system of Hong et al. (2022), we enhance the system's fact-retrieving ability by enriching the model with state chains to retrieve facts dynamically. Besides, this research investigates the correlation between the entailment reasoning types and the rhetorical relations (Mann and Thompson, 1987). On this basis, we train a rhetorical relation classifier and have the TER system learn to imitate the rhetorical relation prediction during the reasoning steps, thus infusing the rhetorical knowledge into the TER system. Furthermore, in order to alleviate the error propagation issue of our TER system in the bottom-up and top-down tree-building process, we explore improving the quality of the generated intermediate conclusions. Specifically, we design the instruction-tuning data tailored for TER, with which we tune the LLMs to more accurately generate the conclusion of each intermediate step. Experimental results on EntailmentBank demonstrate the effectiveness of our proposed approaches. Notably, to our knowledge, this research is the first to investigate the effects of rhetorical knowledge and LLM-driven interpretability on tree-structured entailment reasoning.

2. Related Work

QA explanation. Previous research on QA explanation mainly focuses on searching for facts that can support the hypothesis but neglects the relations inside the retrieved facts (Jansen and Ustalov, 2019; Yadav et al., 2019, 2020; Jhamtani and Clark, 2020). With the recent publication of EntailmentBank (Dalvi et al., 2021), more and more researchers have turned to tree-structured explanation generation. The studies mainly lie in two styles. On the one hand, Dalvi et al. (2021) cast the entailment tree construction process as a sequence-to-sequence generation task and introduced the seq2seq EntailmentWriter based on the T5 transformer (Raffel et al., 2020). On the other hand, more and more researchers cast the task into a multi-step generation process (Ribeiro et al., 2022; Bostrom et al., 2022; Hong et al., 2022; Liu et al., 2022). Among these studies, Ribeiro et al. (2022) propose to generate explanations step by step, where the reasoning and fact-retrieving processes are combined to allow the model to leverage intermediate conclusions for retrieving and reasoning. Bostrom et al. (2022) decompose the original task into separate steps through a search procedure, which has been proven to generate better quality than the end-to-end T5 model. Liu et al. (2022) introduce the reinforcement learning framework into this task and design a flexible reward scheme to consider the entire reasoning chain for better TER. Recently, Hong et al. (2022) transform the TER task into the process of constructing tree nodes in bottom-up or top-down directions step by step, which serves as the baseline system in our work.

Rhetorical discourse structure. Rhetorical structure theory (RST) (Mann and Thompson, 1987) is a fundamental discourse structure theory, which describes each article as a constituency tree. Subsequently, with the publication of RST Discourse Treebank (Carlson et al., 2002), more and more RST parsers have been proposed so far (Zhang et al., 2020, 2021; Yu et al., 2022; Kobayashi et al., 2022), which push RST parsing to a relatively mature stage. From the application viewpoint, many efforts have been made to apply this discourse knowledge to NLP applications like text summarization (Xu et al., 2020), text categorization (Ji and Smith, 2017), and document-level machine translation (Tan et al., 2022). This paper explores applying such discourse knowledge to TER.

LLM-driven explanation generation. With the rapid development of LLMs (Brown et al., 2020; Chung et al., 2022; Touvron et al., 2023a; OpenAI, 2023; Touvron et al., 2023b), LLM-driven explanation generation has become more and more critical, especially when more and more facts have shown that the explanation can help improve the model's understanding of natural language (Wei et al., 2021; Mishra et al., 2022b,a; Parmar et al., 2022). In particular, the recent explanation in a

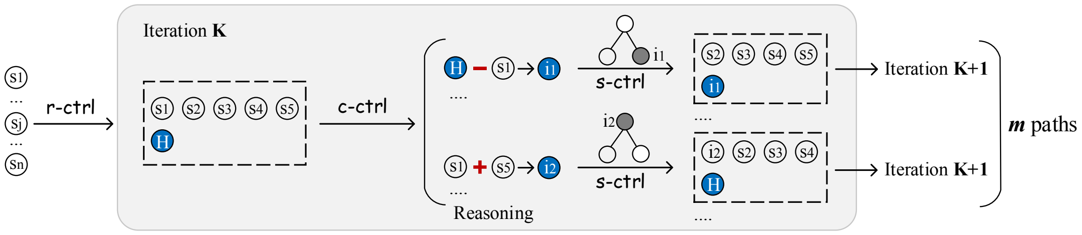


Figure 2: Multi-module entailment tree generation framework. Signs r-ctrl, c-ctrl, and s-ctrl refer to the controller module used for fact retrieval, combination selection, and state selection, respectively.

chain-of-thought fashion (Wei et al., 2022) can wildly dig out the ability of current LLMs thus improving their ability in few-shot logic inference. In this paper, we propose a scheme of tuning and incorporating LLMs tailored for the TER task.

3. Baseline

In this paper, we take the recent module-based entailment tree construction framework METGEN[1] as our baseline system. As shown in Figure 2, METGEN treats the TER task as a multi-step bidirectional reasoning process. Initially, a transformer-based controller (r-ctrl) is used to select facts from all candidates, $s_1, \ldots, s_n$, obtaining the premises, $s_1, \ldots, s_5$, most related to the hypothesis. Then, the five selected premises are further combined in two different forms for top-down abductive (-) and bottom-up deductive (+) reasoning at each iteration. For example, in Figure 2, the hypothesis and the leaf node s_1 are combined to form the low-layer node i_1 from the top down; While the two leaves s_1 and s_5 are combined to form the upper-layer node i_2 from the bottom up. To achieve the process, the controller (c-ctrl) module is first used to select suitable combinations within the premises and hypothesis, then a T5-based generative reasoning module is utilized to draw an intermediate conclusion for each new node. Subsequently, with these newly built internal nodes, some new reasoning states are obtained, and the controller (s-ctrl) module is further utilized to select m good states to form m independent beam paths for subsequent iterations. The TER process ends when the hypothesis is proved, or the reasoning process reaches the maximum iteration setting.

4. Method

Based on the above-mentioned benchmark system, this section introduces our TER system with enhancements on fact retrieval, rhetorical perception, and LLM interpretability integration.

4.1. Fact Retrieval

Fact retrieval, as the first stage of the TER task, is critical since it can cause error propagation to the subsequent entailment tree building process. As depicted in Figure 2, the baseline system establishes a solid framework of bidirectional entailment reasoning, where the controller module is trained to select facts at the initial stage once and for all. Such a scheme may potentially lead to severe error propagation in subsequent reasoning stages (Ribeiro et al., 2022). To mitigate this issue, we develop an approach of training the retrieval controller iteratively throughout the multi-step entailment tree reasoning process, enhancing the overall performance of the retrieval module. For the representation of facts and states, we follow the previous work (Hong et al., 2022) to utilize a pre-trained transformer of `albert-xxlarge-v2` (Lan et al., 2019) to encode each sequential state. For example, in the initial state, `[CLS]H[SEP]s1[SEP]...sn[SEP]`, we use the embedding of `[CLS]` to represent the state and compute the averaged token embeddings within `H` and `sn` to represent the hypothesis and the `n`-th fact, respectively. Different from the baseline system, we propose training the controller to select correct facts at each step while considering the tracked states and the hypothesis, as follows:

$$H_t^{state} = mean(\sum_i^t H_i^{[CLS]})$$

$$\varrho_t \sim \pi_\theta(\boldsymbol{\varrho_t} | H_t^{state}, H^{hypo})$$

where $H_i^{[CLS]}$ refers to the state representation at the i-th reasoning step, H_t^{state} represents the tracked state sequence at the t-th step, and H^{hypo} denotes the embedding of the hypothesis. Subsequently, the controller is trained by maximizing the probability of selecting the "good" fact ϱ_t, while concurrently minimizing the probability of choosing irrelevant or "bad" facts within the current state.

Previous work (Bengio et al., 2009) has substantiated that training with "easy" examples can yield lower generalization error, which is of great reference value for the current need to select premises

[1] https://github.com/Raising-hrx/MetGen

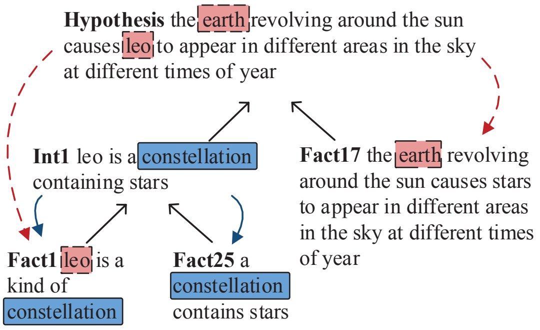

Hypothesis the earth revolving around the sun causes leo to appear in different areas in the sky at different times of year
Int1 leo is a constellation containing stars
Fact17 the earth revolving around the sun causes stars to appear in different areas in the sky at different times of year
Fact1 leo is a kind of constellation
Fact25 a constellation contains stars

Figure 3: Lexical clues within entailment trees.

from many noisy facts. To facilitate "easy" learning in the fact-retrieving process, we define the criteria for the "good" facts with two points: textual intersection and structural correlation.

Textual intersection. Previous work (Ribeiro et al., 2022) has provided evidence that numerous leaf facts, seemingly unrelated to the root, may possess connections with intermediate conclusions. With this empirical basis, we propose training the model to retrieve facts during the internal node generation process. To this end, at timestep t, we consider facts that exhibit a non-stop word intersection with the hypothesis and the intermediate conclusion generated at step $t-1$ as "good" facts to retrieve, thereby increasing the probability of $\hat{p}_t^{pos} = \hat{p}(s_t|s_{1...n}, H^{hypo}, H_{t-1}^{state})$. For the example in Figure 3, considering that only Fact1 and Fact17 have lexical connections (i.e., the words earth and leo) with the hypothesis at the initial stage, we take them as good facts to retrieve, and the negative ones can be randomly selected from the corpus, excluding the ground truth.

Structural correlation. In addition to training the model's ability to discern relevant facts from irrelevant ones, we also introduce a greedy approach to retrieve the most suitable facts at each step, mitigating the risk of falling into local optima. A review of EntailmentBank reveals a linguistic pattern: nouns employed within a tree node τ are more likely to be reused in nodes proximate to τ than those farther away. Drawing inspiration from this, we select the fact that is structurally closest to the newly built internal node at step t as the optimal fact to retrieve at step $t+1$. Furthermore, once τ emerges as the root of a sub-tree following bottom-up deductive reasoning, there is no longer a necessity to retrieve the leaf facts related to τ. Similarly, when τ serves as a child node after the top-down abductive step, there is no need to retrieve its sibling facts anymore. For example, in Figure 3, given that the deductive action is taken at the initial state to combine Fact1 and Fact25 into Int1, then Fact17 is structurally closest to the newly built node and deemed the most suitable fact to retrieve.

To achieve the above goals, we employ a two-fold training strategy that minimizes the max-margin ranking loss between the positive and negative facts, coupled with the negative log-likelihood (NLL) loss, to train the model in selecting the most appropriate fact at each step.

$$L = \sum_{steps} (\lambda_1 max(0, M + \hat{p}_t^{neg} - \hat{p}_t^{pos}) - \lambda_2 log(\hat{p}_t^{opt}|\theta))$$

where parameters λ_1 and λ_2 are used to balance the two training components, $\hat{p}_t^{neg}$ and $\hat{p}_t^{pos}$ denote the retrieval probabilities for negative and positive facts respectively, M denotes the max-margin value, and $\hat{p}_t^{opt}$ means the probability of selecting the optimal fact at step t.

4.2. Rhetorical Relation Incorporation

Statistics in (Dalvi et al., 2021) show that the reasoning types of Substitution, Conjunction, and If-then collectively cover over 90% of the steps in EntailmentBank. Previous paper (Hong et al., 2022) has explored utilizing well-formed synthetic data containing logical regularities of the three reasoning types to train their TER system. Building upon the examples used in (Hong et al., 2022), we delve deeper into the correlation between reasoning types and rhetorical relations, as illustrated in Table 1. Examining the tree structure encompassing s_1, s_2, and i_1, the substitution type is employed to transform the entity "planet" in s_1 into "earth" in s_2, ultimately leading to the conclusion i_1. In this case, we designate s_2 as the nucleus unit in RST, with s_1 serving as the satellite unit responsible for elaborating upon the nucleus unit on "the earth planet". Therefore, the rhetorical relation of Elaboration can be bridged between the two premises. Analogically, the conjunction reasoning type takes the two facts s_3 and s_4 as two separate descriptions of "chemical splashing", where we can use the rhetorical relation of Joint to combine the two facts to reach the conclusion. Moreover, in cases involving the if-then reasoning type, the fact s_5 plays the role of a common logical premise, with the fact s_6 acting as a specific instance to reach the intermediate conclusion i_3. In this case, s_5 assumes a rhetorical Background role. These examples underscore the profound correlation between reasoning types and rhetorical relations, prompting us to integrate such knowledge into our TER system.

To construct a suitable training dataset for rhetorical relations, we extract sub-trees from the vanilla RST-DT corpus (Carlson et al., 2002), yielding a total of 4,629 instances for training, 491 for vali-

Example	Reasoning	Rhetorical
s_1: the mass of a **planet** causes the pull of gravity on that **planet**. s_2: `earth` is a kind of **planet**. i_1: the mass of `earth` causes the pull of gravity on `earth`.	$s_1\&s_2 \xrightarrow{Subs.} i_1$	$s_1 \overset{Elab.}{\rightleftarrows} s_2$
s_3:chemical splashing can cause **harm to humans/to the eyes**. s_4:`chemical splashing` sometimes occurs `during experiments`. i_2:`chemical splashing during experiments` can cause **harm to eyes**.	$s_3\&s_4 \xrightarrow{Conj.} i_2$	$s_3 \overset{Joint}{\rightleftarrows} s_4$
s_5: if **something requires something else** then that `something else is important to that something`. s_6: **nuclear fusion is required for star formation.** i_3: `nuclear fusion is important to star formation.`	$s_5\&s_6 \xrightarrow{If-then} i_3$	$s_5 \overset{Back.}{\rightleftarrows} s_6$

Table 1: Correlation between the three entailment reasoning types and the rhetorical discourse relations.

dation, and 604 for testing.[2] Each instance comprises three components: the left-node text, the right-node text, and the rhetorical relation between the two nodes. Subsequently, we calculate the relation prediction score as below.

$$G_{rhe} = \ell(W_{rhe} \times (o_{node\_L} \oplus o_{node\_R}))$$

where ℓ denotes the log-softmax function, $\oplus$ signifies vector concatenation, and o_{node_L} and o_{node_R} means the representation of the two textual tree nodes obtained from the pre-trained XLNet (Yang et al., 2019). Then the rhetorical relation classifier is trained by minimizing the NLL loss of predicting the correct rhetorical relations. In order to incorporate such rhetorical knowledge into the entailment parser, when combining two premises through a bottom-up deductive step, we have the pre-trained rhetorical classifier as the teacher to generate the rhetorical relation ($\widetilde{r}$) between the two premises. Then, we have the TER system to imitate such knowledge by training it to predict as the pre-trained classifier.

$$\ell(FNN_{imt}(H_{si}, H_{sj})) \sim \widetilde{r}$$

where H_{si} and H_{sj} denote the representation of the tree nodes to combine (see Subsection 4.1).

4.3. LLM-driven Intermediate Conclusion Generation

Recent years have seen the impressive performance of LLMs (Brown et al., 2020; Chung et al., 2022; OpenAI, 2023) in NLP tasks including explanation generation (Li et al., 2022). Although previous work has shown the remarkable ability of LLMs on natural language reasoning in a chain-of-thought style (Wei et al., 2022), the reasoning within an entailment tree framework remains an ongoing area of research. It is worth mentioning that our investigation on `gpt-3.5-turbo` shows two noteworthy tendencies when LLMs are tasked with one-shot learning in the context of TER. On one hand, LLMs, given their substantial knowledge base, often expedite the process of reaching the hypothesis based on the provided facts. Consequently, they tend to generate relatively shallow and wide entailment trees. On the other hand, the premises within each entailment tree are usually organized level by level from basic inferences to complicated ones, but LLMs usually do not follow such an easy-to-hard rule. Since they are trained as perfect text sequence generators, they prefer explaining relations within facts using a more coherent and flexible text sequence instead. Nevertheless, the impressive generation capability of current LLMs is definitely an opportunity for intermediate conclusion generation in TER.

Similar to many tree parsing tasks, the building of entailment trees also suffers from the error propagation issue. Specifically, in TER, the lower-level low-quality intermediate conclusion of each tree node built in the deductive mode will propagate errors upward, while the upper-level low-quality conclusions built in the abductive mode will propagate errors downward. Since the quality of an entailment tree highly depends on the intermediate conclusions generated step by step, we speculate that employing the interpretability of LLMs for intermediate conclusion generation will help alleviate the error propagation issue.

To this end, we propose an instruction-tuning scheme that encompasses two distinct styles of instruction-tuning data. First, we build the instruction prompt with the reasoning mode indicator `abductive mode`, the hypothesis, and the gold standard premises considered. We tune the model to generate an intermediate conclusion that serves as one of the child nodes within the newly built sub-tree. Second, we form the instruction prompt with the reasoning mode indicator `deductive`

[2] The vanilla RST trees contain 55.6 leaves on average, and the entailment trees contain 4.4 on average. Therefore, we only consider the RST sub-trees with a depth of less than 4 in this research. We consider tree nodes associated with relations such as `Elaboration`, `Joint`, and `Background`, as they are most closely aligned with the reasoning types in TER. The textual representation of each internal node is achieved by concatenating the leaves within the node map.

mode and the known premises considered, and then instruct the model to generate an intermediate conclusion that assumes the root node of the newly built sub-tree. Drawing from the training set of EntailmentBank, we construct a total of 15,719 tuning instances, partitioned into 13,741 for training and 1,978 for validation. Leveraging this dataset, we fine-tune the foundation model Flan_T5_XL(3B) (Chung et al., 2022) for three epochs. Subsequently, the tuned model serves as the reasoning module within our TER system.

5. Experimentation

Following previous research on TER, we conduct experiments on EntailmentBank (Dalvi et al., 2021), which is known as the pioneering corpus tailored for science QA explanations in the form of entailment trees. The corpus contains a total of 1,840 entailment trees, with 1,313 allocated for training, 187 for validation, and 340 for testing. Consistent with Dalvi et al. (2021), we consider three tasks with increasing difficulties: Task1 with only gold standard facts, Task2 with gold facts and 15-20 distractors for retrieving, and Task3 with facts from the full corpus for retrieving.

5.1. Experimental Settings

Evaluation metric. In line with prior evaluation metrics (Dalvi et al., 2021; Hong et al., 2022), we evaluate the quality of predicted entailment trees post-alignment by examining three key aspects: (1) The score for Leave facts detection, determined by comparing the leaves of the predicted trees with the gold standard. (2) The score for structural Steps, gauged by if the child nodes of internal nodes match the ground truth. (3) The score for Intermediate conclusions, obtained by comparing the aligned intermediate conclusions of predicted nodes with those of the gold standard.[3]

N-ary to binary tree conversion. To the best of our knowledge, prior studies (Dalvi et al., 2021; Hong et al., 2022) have provided insights into binary entailment trees, suggesting that "n-premise steps (n>2) could be further decomposed into several valid 2-premise steps". Inspired by this, we propose a straightforward left-branching scheme aimed at converting non-binary trees into binary structures, as depicted in Figure 4. In this scheme, for root nodes with multiple children, we combine the leftmost two children into a new intermediate node Int1. The newly created node then serves as a child node of the root. This process continues until all internal nodes contain only two children, obtaining a binary entailment tree. Furthermore, to avoid misleading the reasoning process and minimize the need for human annotation, we obtain the conclusion for each newly built internal node by combining the facts of its child nodes.

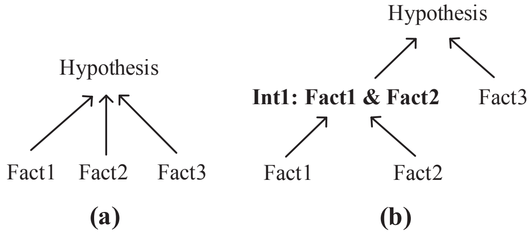


Figure 4: Convert n-ary to binary entailment tree.

Rhetorical relation detection. As stated before, this paper explores the effects of RST knowledge on TER. We experiment on two pre-trained language models, i.e., RoBERTa-large (Liu et al., 2019) and XLNet (Yang et al., 2019), which have been proven effective in RST discourse analysis (Kobayashi et al., 2022). The performance of the two systems on our built test corpus is 74.1% for RoBERTa and 79.3% for XLNet on accuracy. Finally, we employ the classifier trained on XLNet to predict rhetorical relations for the TER system to imitate.

5.2. Experimental Results

We draw upon results from two closely related papers to our research for performance comparison.

- **EntailmentWriter** (Dalvi et al., 2021). The first sequence-to-sequence entailment tree generator in TER directly produces linearized trees using an encoder-decoder framework, based on the hypothesis, QA pair, and candidate facts.
- **Baseline** (Hong et al., 2022). A module-based entailment reasoner that splits the once-for-all sequence-to-sequence reasoning process into a multi-step reasoning process. Notably, it is the first system to consider entailment reasoning in both bottom-up and top-down directions, and it serves as our primary baseline system for performance comparison.

Table 2 reports the primary results. In each task, the first two rows show the performance of previous systems under similar settings[4]. Meanwhile,

[3] We report both F1 and AllCorrect (ALCC) scores as performance. For leaves, steps, and intermediates, an ALCC score of 1 signifies that all predicted leaves, steps, or intermediate conclusions within an entailment tree are correct. And the strict Overall ALCC metric is used to assess whether all leaves, steps, and intermediates of the entire tree are correct.

[4] Both the baseline systems and ours employ the same T5-large transformer for reasoning and use the

	Method	Leaves		Steps		Intermediates		Overall
		F1	ALCC	F1	ALCC	F1	ALCC	ALCC
Task1	EntailmentWriter (Dalvi et al., 2021)	98.4	84.1	50.0	38.5	67.0	35.9	34.4
	Baseline (Hong et al., 2022)	**100**	**100**	**57.7**	41.9	**70.8**	39.2	36.5
	Baseline + Rhetorical	**100**	**100**	56.9	42.1	68.8	37.1	34.7
	Baseline + Rhetorical ♠	**100**	**100**	57.6	**42.9**	**70.8**	**40.0**	**36.8**
Task2	EntailmentWriter (Dalvi et al., 2021)	83.2	35.0	39.5	24.7	62.2	28.2	23.2
	Baseline (Hong et al., 2022)	82.7	46.1	41.3	29.6	61.4	32.4	27.7
	Baseline + Retrieving	83.4	47.9	42.2	29.4	62.5	**32.6**	27.4
	Baseline + Rhetorical	83.3	**49.7**	42.1	30.3	61.0	31.8	**28.2**
	Baseline + Both	83.5	48.8	40.5	28.8	62.4	30.9	26.2
	Baseline + Both ♠	**84.8**	49.4	**42.9**	**30.9**	**62.9**	32.1	27.1
Task3	EntailmentWriter (Dalvi et al., 2021)	30.9	1.2	4.4	1.2	28.8	5.6	1.2
	Baseline (Hong et al., 2022)	34.8	8.7	9.8	8.6	36.6	**20.4**	8.6
	Baseline + Retrieving	36.4	**10.0**	**11.7**	**10.0**	36.5	20.3	**10.0**
	Baseline + Rhetorical	36.8	8.5	10.6	8.5	**37.7**	20.3	8.5
	Baseline + Both	**36.9**	**10.0**	11.6	**10.0**	37.1	20.3	**10.0**
	Baseline + Both ♠	36.8	**10.0**	11.6	**10.0**	37.0	20.3	**10.0**

Table 2: Main TER results of the proposed approaches. Underline denotes the score outperforms the baseline system, the scores in bold yield the best among all. Sign ♠ denotes the LLM-based system.

the remaining rows ("Baseline+X") present a series of ablation studies. Note that since Task 1 already incorporates gold standard facts as inputs, applying our retrieval enhancement method would be meaningless. Thus we only augment the TER model's retrieval ability for Tasks 2 and 3. We draw the following observations and conclusions.

Enhancement by fact retrieval. A comparison between "Baseline + Retrieving" and "Baseline" reveals significant improvements in fact retrieval performance. Specifically, the designed scheme of training the TER model to distinguish between good and bad facts and selecting the best fact at each reasoning step substantially improves the results on `Leaves`. Besides, the improvements in fact retrieval are observed to cascade into the enhancements in `Steps`, `Intermediates`, and `Overall` performance to different extents.

Enhancement by rhetorical perception. The results in Task 1 show that enhancing the controller model with rhetorical relation detection capabilities results in a notable reduction in `Intermediates` and `Overall`. However, in Tasks 2 and 3, where noisy facts are introduced into the reasoning process, our method's advantage emerges clearly, even when both retrieval and rhetorical enhancements are applied. These findings suggest that infusing rhetorical discourse knowledge into TER enhances its adaptability to real-world scenarios with noisy facts. Additionally, we observe that simultaneously applying both retrieval and rhetorical enhancements does not consistently yield the best results. We speculate that the two types of information sometimes complement each other while in other cases be incompatible, leading to significant variations in results when applied together.

	Method	Leaves	Steps	Int.	Overall
Task1	`LLaMa`	**100**	52.3	47.2	23.2
	`Alpaca`	**100**	53.5	53.0	26.2
	`FLT-XL`	**100**	**57.6**	**70.8**	**36.8**
Task2	`LLaMa`	75.3	35.5	43.4	19.7
	`Alpaca`	76.7	35.5	48.6	20.6
	`FLT-XL`	**84.8**	**42.9**	**62.9**	**27.1**
Task3	`LLaMa`	36.6	11.2	36.6	9.8
	`Alpaca`	**36.8**	11.2	36.8	**10.0**
	`FLT-XL`	**36.8**	**11.6**	**37.0**	**10.0**

Table 3: TER results (F1) on different LLMs.

Enhancement by LLM integration. When applying the tuned LLM (♠) to our system for intermediate conclusion generation, we observe substantial improvements in `Steps` and `Intermediates` for Tasks 1 and 2. It indicates that leveraging LLM can enhance the generation of accurate intermediate conclusions and improve entailment tree construction. However, the effects of LLM in Task 3 appear to be limited. An examination of the predicted trees reveals that most trees in Task 3 consist of no more than three leaves, requiring fewer intermediate conclusions compared to the other two tasks, which reduces LLM's impact. Notably, fact retrieval remains a prominent challenge in TER.

5.3. Comparison of Foundation Models

Table 3 presents the performance across different foundation models. It shows that employing `LLaMa(7B)` yields the worst performance. However, when we initially tune `LLaMa` using the 52K

same `prefixed` setting for the controller module. As detailed in (Hong et al., 2022), the `prefixed` setting employs the minor model parameters to adapt to the cases with different reasoning types and directions.

	Data	Leaves		Steps		Intermediates		Overall
		F1	ALCC	F1	ALCC	F1	ALCC	ALCC
Task1	Multifurcating Tree	100	100	57.6	42.9	**70.8**	**40.0**	36.8
	Binary Tree	100	100	**60.0**	**45.9**	68.3	37.6	**37.6**
Task2	Multifurcating Tree	84.8	49.4	42.9	30.9	**62.9**	**32.1**	27.1
	Binary Tree	84.8	49.4	**44.5**	**31.8**	59.3	28.8	27.1
Task3	Multifurcating Tree	36.8	10.0	11.6	10.0	**37.0**	**20.3**	10.0
	Binary Tree	36.8	10.0	11.6	10.0	34.1	18.5	10.0

Table 4: Our results on the multifurcating and binary entailment trees.

alpaca data[5] and subsequently fine-tune it with our data, a significant performance improvement is observed. It indicates that exposure to various tasks and our instruction-tuning data can aid `LLaMa` in better understanding the TER task. Moreover, the overall results show that `Flan_T5_XL(3B)` (abbr., `FLT-XL`) consistently outperforms the others. To delve deeper into these findings, we conduct a thorough review of system outputs and observe that the conclusions generated by `FLT-XL` are concise and exhibit a similar style to those produced by humans. In contrast, `LLaMa` tends to generate overly verbose conclusions, resulting in a significant performance decline for style mismatches. Similar to the results in Table 2, the impact of different LLMs on Task 3 remains indiscernible, which is primarily due to the formidable challenge of fact retrieval.

5.4. Results on Binary Entailment Trees

Here we report the performance of our system on binary entailment trees, as shown in Table 4. In the first two tasks, where the fact retrieval is less challenging and more internal nodes are built, the scores on `Steps` go up, and that on `Intermediates` go down for all three tasks. Since our system is trained to generate a binary structure at each step, which is more compatible with the binary data, making the scores on the structural `Steps` higher. Besides, as stated before, when converting the n-ary trees, we get the intermediate conclusions of new nodes by combining its leaf nodes' facts, which is inconsistent with the composition principle of conclusions in EntailmentBank, resulting in performance degradation on `Intermediates`.

In general, we argue the left-branch binarization scheme may have two benefits: On one hand, taking "node merging" as a special reasoning type to build the binary structure without losing information can reduce the complexity of the vanilla n-ary trees. On the other hand, this paper promotes the research on binary entailment trees, aiming to provide ideas for subsequent LLM-based TER, that is, using "binary tree" to restrict foundation models to reason under the easy-to-hard rule, thereby generating a linearized binary entailment tree.

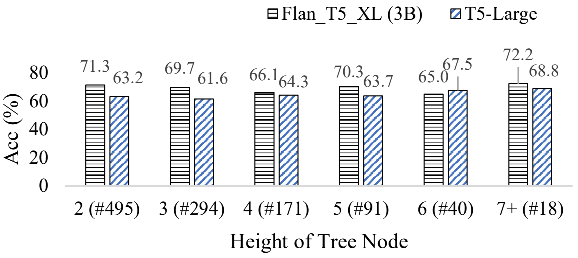


Figure 5: Intermediate conclusion generation accuracy of tree nodes with different heights.

5.5. Analysis on Error Propagation

In NLP tasks that involve constituency tree construction, the quality of upper-layer nodes significantly impacts the lower-layer nodes in the bottom-up TER process, and vice versa in the top-down process. To these dependencies, we argue that the misprediction of intermediate conclusions could lead to substantial error propagation issues within TER. In this work, error propagation issues can manifest in two scenarios: error propagation from bottom to up in the deductive mode and error propagation from top to down in the abductive mode. Figure 5 illustrates the accuracy of intermediate conclusion generation with respect to entailment tree nodes with different heights. It shows that the TER results on low- and high-level entailment tree nodes tend to yield higher scores, this trend is more obvious when using the tuned `FLT-XL` for experimentation. It indicates that the intermediate conclusion generation for low- and high-layer tree nodes is less susceptible to the error propagation issue in this bidirectional TER system, while mid-height nodes are more affected. Besides, `FLT-XL`'s superior performance on low- and top-level nodes reduces errors to a certain extent, enhancing its mid-height conclusion generation.

6. Conclusion

This paper focuses on tree-structured entailment reasoning for science QA with three critical aspects.

[5] github.com/tatsu-lab/stanford_alpaca

Firstly, we strengthen the fact-retrieving capability of the TER model through dynamic reasoning training. Secondly, we explore the correlation between reasoning types and rhetorical relations, harnessing RST knowledge to augment the system's adaptability. Finally, we design task-specific tuning data, coupled with the LLM integration, which further bolsters the system's TER performance. Experimental results on the EntailmentBank corpus demonstrate the effectiveness of our proposed approaches. Codes and the binarized Entailment tree data will be available upon email request.

As described in the paper, we encountered performance bottlenecks in this research primarily due to insufficient fact retrieval quality, so investigating the potential of utilizing LLMs for fact retrieval would be a focal point in our future research. Furthermore, as outlined in the paper, this study contributes the binarized Entailment data, aiming to advance LLM-based Entailment Reasoning in future work, specifically by constraining the foundational models to engage in reasoning following the easy-to-hard rule to generate linearized trees.

7. Limitations

We summarize the limitations of this paper for subsequent researchers to refer to and boost the development of TER. First, all the methods in our paper were tested essentially using automatic proxy metrics rather than manual evaluation. The automatic evaluation method is far from perfect when facing this novel and challenging TER task, and the exploration of manual evaluation is urgently needed. Secondly, this paper aims to introduce novel and creative methodologies, but the performance gains are moderate. Thirdly, the baseline framework is innovative but already complex, and our research further introduces the combination of various techniques, making the experimental settings complex and hard to follow and apply to real scenarios. Moreover, the proposed method enhances the model's fact retrieval ability through multi-step fact retrieving learning, but an application for the inference stage remains challenging in this complex baseline framework. We are committed to tackling the aforementioned challenges in future work endeavors.

8. Acknowledgments

This research was supported by the funding (CR-2021-001) from the Institute for Infocomm Research (I^{2}R) under A*STAR ARES, Singapore. We thank Zhuohan Liu, Tharini De Silva, and Liangming Pan for their insightful discussions. We express our gratitude to the three anonymous reviewers who highly rated this paper (4.33 on average) and offered insightful comments that significantly contributed to the enhancement of this work.

9. References

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of ICML*, pages 41–48.

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2011. The seventh pascal recognizing textual entailment challenge. In *TAC*.

Kaj Bostrom, Zayne Sprague, Swarat Chaudhuri, and Greg Durrett. 2022. Natural language deduction through search over statement compositions. *arXiv preprint arXiv:2201.06028*.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Lynn Carlson, Mary Ellen Okurowski, and Daniel Marcu. 2002. *RST discourse treebank*. Linguistic Data Consortium, University of Pennsylvania.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

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

Bhavana Dalvi, Peter Jansen, Oyvind Tafjord, Zhengnan Xie, Hannah Smith, Leighanna Pipatanangkura, and Peter Clark. 2021. Explaining answers with entailment trees. *arXiv preprint arXiv:2104.08661*.

Ruixin Hong, Hongming Zhang, Xintong Yu, and Changshui Zhang. 2022. METGEN: A module-based entailment tree generation framework for answer explanation. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1887–1905, Seattle, United States. Association for Computational Linguistics.

Peter Jansen and Dmitry Ustalov. 2019. TextGraphs 2019 shared task on multi-hop inference for explanation regeneration. In *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing*, pages 63–77, Hong Kong. Association for Computational Linguistics.

Peter A Jansen, Elizabeth Wainwright, Steven Marmorstein, and Clayton T Morrison. 2018. Worldtree: A corpus of explanation graphs for elementary science questions supporting multi-hop inference. *arXiv preprint arXiv:1802.03052*.

Harsh Jhamtani and Peter Clark. 2020. Learning to explain: Datasets and models for identifying valid reasoning chains in multihop question-answering. In *Proceedings of EMNLP*, pages 137–150, Online. Association for Computational Linguistics.

Yangfeng Ji and Noah A. Smith. 2017. Neural discourse structure for text categorization. In *Proceedings of ACL*, pages 996–1005, Vancouver, Canada. Association for Computational Linguistics.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. Qasc: A dataset for question answering via sentence composition. In *Proceedings of AAAI*, volume 34, pages 8082–8090.

Naoki Kobayashi, Tsutomu Hirao, Hidetaka Kamigaito, Manabu Okumura, and Masaaki Nagata. 2022. A simple and strong baseline for end-to-end neural RST-style discourse parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6725–6737, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Shiyang Li, Jianshu Chen, Yelong Shen, Zhiyu Chen, Xinlu Zhang, Zekun Li, Hong Wang, Jing Qian, Baolin Peng, Yi Mao, et al. 2022. Explanations from large language models make small reasoners better. *arXiv preprint arXiv:2210.06726*.

Tengxiao Liu, Qipeng Guo, Xiangkun Hu, Yue Zhang, Xipeng Qiu, and Zheng Zhang. 2022. Rlet: A reinforcement learning based approach for explainable qa with entailment trees. *arXiv preprint arXiv:2210.17095*.

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

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521.

William C Mann and Sandra A Thompson. 1987. *Rhetorical structure theory: A theory of text organization*. University of Southern California, Information Sciences Institute Los Angeles.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. 2022a. Reframing instructional prompts to GPTk's language. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 589–612, Dublin, Ireland. Association for Computational Linguistics.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022b. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of ACL*, pages 3470–3487, Dublin, Ireland. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report.

Mihir Parmar, Swaroop Mishra, Mirali Purohit, Man Luo, Murad Mohammad, and Chitta Baral. 2022. In-BoXBART: Get instructions into biomedical multi-task learning. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 112–128, Seattle, United States. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Danilo Ribeiro, Shen Wang, Xiaofei Ma, Rui Dong, Xiaokai Wei, Henry Zhu, Xinchi Chen, Zhiheng Huang, Peng Xu, Andrew Arnold, et al.

2022. Entailment tree explanations via iterative retrieval-generation reasoner. *arXiv preprint arXiv:2205.09224*.

Oyvind Tafjord, Bhavana Dalvi Mishra, and Peter Clark. 2020. Proofwriter: Generating implications, proofs, and abductive statements over natural language. *arXiv preprint arXiv:2012.13048*.

Xin Tan, Longyin Zhang, Fang Kong, and Guodong Zhou. 2022. Towards discourse-aware document-level neural machine translation. In *Proc. 31st Int. Joint Conf. Artificial Intelligence, Vienna, Austria*, pages 4383–4389.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Discourse-aware neural extractive text summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5021–5031, Online. Association for Computational Linguistics.

Vikas Yadav, Steven Bethard, and Mihai Surdeanu. 2019. Quick and (not so) dirty: Unsupervised selection of justification sentences for multi-hop question answering. In *Proceedings of EMNLP-IJCNLP*, pages 2578–2589, Hong Kong, China. Association for Computational Linguistics.

Vikas Yadav, Steven Bethard, and Mihai Surdeanu. 2020. Unsupervised alignment-based iterative evidence retrieval for multi-hop question answering. In *Proceedings of ACL*, pages 4514–4525, Online. Association for Computational Linguistics.

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*, volume 32. Curran Associates, Inc.

Nan Yu, Meishan Zhang, Guohong Fu, and Min Zhang. 2022. RST discourse parsing with second-stage EDU-level pre-training. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4269–4280, Dublin, Ireland. Association for Computational Linguistics.

Longyin Zhang, Fang Kong, and Guodong Zhou. 2020. Syntax-guided sequence to sequence modeling for discourse segmentation. In *Natural Language Processing and Chinese Computing*, pages 95–107, Cham. Springer International Publishing.

Longyin Zhang, Fang Kong, and Guodong Zhou. 2021. Adversarial learning for discourse rhetorical structure parsing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 3946–3957, Online. Association for Computational Linguistics.

10. Appendices

Appendix A. System Settings Following the baseline system (METGEN) (Hong et al., 2022), we conducted experiments on EntailmentBank, 1,313 for training, 187 for validation, and 340 for testing. We considered three tasks with increasing difficulties, i.e., Task1 based on gold standard facts, Task2 with gold facts and 15-20 distractors for retrieving, and Task3 with facts from the full corpus for retrieving. To strengthen METGEN and retain its advantages, based on their published code set, we shape our model training process in a multi-round fashion. Specifically, we first borrowed Hong et al. (2022)'s model as the base model, then at each round, we first tuned the controller to enhance its ability in both fact retrieval and rhetorical relation prediction for 10 epochs. Then we further tuned the model as (Hong et al., 2022) for 10 epochs to avoid parameter confusion. We trained the system for 5 rounds to reach the final system. All system settings are inherited from (Hong et al., 2022). Notably, for the two parameters λ_1 and λ_2 in Subsection 4.1, we conducted ablation experiments and found that the convergence speed of the two terms has a weak impact on the performance, so we set both to 0.5 to ensure uniform convergence speed for both terms.