On the Progress of Machine Logical Reasoning

1st Qiutong Men
New York University Shanghai
Shanghai, China
qm2017@nyu.edu

2nd Li Yi
New York University Shanghai
Shanghai, China
ly1387@nyu.edu

3rd Xinyu Li
New York University Shanghai
Shanghai, China
xl3133@nyu.edu

***Abstract*—Machine reading comprehension is one of the fundamental tasks of natural language understanding. For span-based question answering, state-of-the-art models based on transformer architecture has surpassed human performance on datasets like SQuAD2.0, etc. However, questions that entail strong capability of logical reasoning remain to be a tough challenge for machines, like those frequently appear in examinations like the Graduate Record Examinations (GRE) and the Law School Admission Test (LSAT). Choices in these questions look similar but contain subtle differences in logic that have a substantial influence on the actual meaning of each sentence. In fact, these logical details can be difficult even for untrained human readers. In this paper, we establish a transformer baseline and a graph model based on the (grammar) dependency graph. Then, we study several state-of-the-art methods and propose a few improvements upon their existing architectures.**

***Index Terms*—Natural Language Processing, Machine Logical Reasoning, Graph Convolutional Networks, Contrastive Learning**

I. Introduction

The primary method to test the level at which one master a language is through reading comprehension questions. This applies to machines as well, with reading comprehension being a major discipline of natural language processing. Current transformer architectures [1] could surpass human performance on benchmark datasets like SQuAD2.0 [2]. However, reading comprehension that requires strong ability of logical reasoning remains a difficulty for machines. Often, these questions appear as multiple choices, while each choice only differ in their logic, having very similar appearance. One must be equipped with the concepts of necessary condition and sufficient condition, the truth table of logical conjuntion, disjunction, implication, etc., and other logic knowledge, to solve these questions. It is no exaggeration to say that compared with span-based question answering, these questions can be extremely harder even for humans, especially untrained readers. One of the benchmark datasets of machine logical reasoning is the ReClor dataset [3].

II. Related Works

ReClor [3] is a dataset proposed by Yu et. al. focusing on complicated machine logical reasoning tasks. The powerful BERT-related baseline model can only achieve about 40 percents accuracy. In the recent 2 years from 2020, there have been surprisingly improvements on this task, and we choose to study the Graph approaches to this task, since logical reasoning is intuitively a task on logical graphs, which gives more intepretability to the models. There are several papers on the leader-board adopts graphical neural network models, including the MERIt group [4] and the DAGN group [5].

A. DAGN

Huang et al. proposed the Discourse-Aware Graph Network (DAGN) approach for logical reasoning [5]. Their philosophy is to do discourse feature enhancement based on the original BERT-related module's output. They proposed to construct logic graphs from the context-question text by using discourse relations as edges and elementary discourse units (EDUs) as nodes. Based on the graph, they calculated the EDU embeddings from the self-defined node embeddings and the message representations. The obtained discourse features are then added with the original token embeddings to enhance the model's performance on logical reasoning. Their result shows that the test accuracy is significantly improved by 2.7%, compared with the RoBERTa backbone. More noteworthy is that the main accuracy improvement occurs in the Test-H set, which proves their approach is able pay more attention to logical relationships. A detailed description of the Test-H set will be provided in Section III.

However, we still observe a few potential improvements regarding the original implementation, including the construction of the EDU embeddings and the structure of the final answer prediction module. A detailed methodology will be provided in Section IV (C).

B. MERIt

Jiao et al. proposed Meta-Path Guided Contrastive Learning for Logical Reasoning (MERIt) [4], which achieves the current SOTA performance. Unlike DAGN, MERIt chooses to further pre-train the current large-scale language models directly to enhance the performance of pre-trained models on logical reasoning. Their approach is to utilize contrastive learning to further train the model, where the positive and negative samples are generated from the Wikipedia corpus [6], based on the entity relations within and among sentences (meta-path). Their result shows that MERIt can significantly improve the test accuracy by 4% (RoBERTa-based [7]). The ALBERT-based [8] MERIt with prompt tuning can achieve a total accuracy of 72.2%, which is very impressive.

The MERIt model is highly integrated and compact, the pre-train process is also hard to reproduce because the GitHub repository does not provide the Wikipedia corpus with entity

TABLE I
STATISTICS OF THE RECLOR DATASET

Feature	ReClor
Num train samples	4638
Num val samples	500
Num test samples	1000
Vocab size	26576
Context length	73.6
Question length	17.0
Option length	20.6

extracted, but its nature of doing pre-training enables the MERIt model to fit in other fine-tuning approaches easily. Section IV (D) will show how we further utilize the pre-trained parameters in our approach.

III. Dataset

The ReClor dataset [3], proposed by Yu et al., is used in this research. This dataset contains 4638 training samples, 500 validation samples, and 1000 testing samples. Each sample is constructed by a context feature, a question feature, and an answers feature, which is a list containing 4 possible options. An extra label feature indicates the correct option for the training set and the validation set. The label of the test set is not provided, but teams can submit predictions on *EvalAI* [9] to view the test accuracy on different subsets. Table 1 lists some other statistics of the ReClor dataset.

The dataset also provides a label to indicate the question type of logical reasoning, for instance, strengthen, weaken, implication, etc. The feature can be useful when analyzing the performance of the model in specific question types.

The test set splits into the Test-E set and the Test-H set. Considering that biases prevalently exist in human-annotated datasets, Yu et al. perform an analysis of the lexical choice and the sentence length on the correct options and the wrong options respectively, and notice there is a slight difference in distribution. This means models might be able to predict the right answer without even knowing the context and the question. To handle this issue, Yu et al. fed the options into several baseline models, including RoBERTa, GPT-2, etc., and picked out 440 biased samples, for instance, predicted correctly several times, to form the Test-E set, while the rest forms the Test-H set. Clearly, the accuracy on the Test-H set is more persuasive to show the model's power in logical inferencing.

IV. Approaches

In this section, we discuss several approaches to tackle this task. They include a baseline model, a model based on relational graph convolutional network [10], two state-of-the-art methods and our modifications upon their existing architectures.

A. Baseline

The baseline of this task is simply a plain classification. Rigorously, for a problem, we are given a context C, a question Q, and 4 choices c_1, c_2, c_3, c_4. We concatenate the context, the question and a choice as a paragraph $P_i = C||Q||c_i$ for $i = 1, 2, 3, 4$. Then, feed each paragraph P_i into a DistilBERT model [11], project the first index of the last hidden state (CLS) to a score s_i. Applying the Softmax function to the scores $S = [s_1, s_2, s_3, s_4]^T$, we get a probability distribution and then minimize the negative log-likelihood loss between that distribution and the ground truth label of this question.

B. DepReasoner

One straightforward idea to enhance the logical inference ability is to explicitly tell the model the grammar dependency of sentences. For each P_i, instead of directly taking out the first index of the last hidden states as a representation of the whole paragraph, we embed the entire last hidden states into the (grammar) dependency graph of the paragraph, using a industrial NLP package SpaCy [12]. An example is shown in Fig. 1, as a dependency graph of a paragraph consisting of two sentences. Specifically, we build the dependency graph in a heterogeneous manner. That is to say, the edges are in different types, each type corresponding to a type of grammar dependency. Besides, the nodes have two features. One is the ”semantic” feature, which is the last hidden states of the transformer encoder. The other one is the part-of-speech (POS) embedding, where we embed each POS type into a vector of length 64. Concatenating these 2 features of the nodes, we get the node features of the graph. Then, by several relational graph convolutional layers [10], we can better aggregate the logic flow of the paragraph. Lastly, for the activated output of the last relational graph convolutional layer, we apply a global attention pooling to get the score s_i, as proposed in the output model of the gated graph sequence neural networks [13].

C. DAGN

DAGN [5] introduces a way of splitting texts into blocks and the corresponding representation of the blocks, called Elementary Discourse Unit(EDU) and EDU Embeddings $\mathbf{e}_n$, where

$$\mathbf{e}_n = \sum_{l \in S_n} \mathbf{t}_l \tag{1}$$

, and $\{\mathbf{t}_l\}$ is defined as the embeddings of tokens in block S_n.

Upon getting the EDU Embeddings, DAGN [5] do the graph reasoning, considering adjacent EDU's information using a linear projection and activation, and adding it to the EDU Embedding. Then the model comes to the downstream part. As DAGN constructs a chain-like graph, it can be feed into a sequential layer, for which the original paper uses a 2-layer Residual Bidirectional GRU. After that, the processed values are concatenated (consisting of information from 4 context-question-option chains) and feed to an FFN layer, which outputs scores for 4 choices respectively. Then by applying

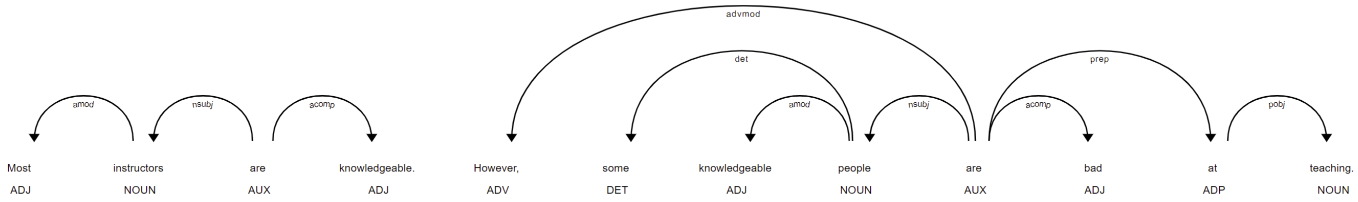

Fig. 1. Example of a dependency graph.

softmax we can to prediction and calculate Cross-Entropy Loss.

There are several potential improvements regarding the original implementation of DAGN, and we implement a proportion of them due to time limits.

1) **EDU Embedding** Original paper uses direct summation of token embeddings to form an EDU's embedding [5], which may cause information loss and is lack of interpretability, because for two equal $\mathbf{e}_i = \mathbf{e}_j$, the model cannot tell the difference between their respective token sets. It may then affect the graph reasoning part. Therefore, we proposed to improve EDU Embedding by using concatenation of token embeddings or apply positional embedding of EDU number on each token embedding.
2) **BiGRU Block** The original paper construct a chain as the graph representation of texts [5], which neglects the possibility of global cross reference. Since GRU is considered relatively hard to pass information across the sequence, we proposed to replace it by BiLSTM or Transformer Encoder [1] and implemented corresponding modules. The modification is shown in Figure. 2 (3).
3) **Edge Representation** Original paper proposes a general projection term when including adjacent nodes' information $W^{r_{ij}} + b_{r_{ij}}$ [5], where r_{ij} denotes the connectivity between two neighboring nodes i, j, regardless the type of connection (supportive, negative, reasoning, enumerating, etc.) We propose to modify the graph reasoning module combining the embeddings of logical separators (e.g. embedding of "because", "such as", "however", etc) with original linear projection, to feed more information into the model.

D. MERIt

MERIt [4] further pre-train the current large-scale language model. They first extracted entities in the Wikipedia corpus [6], and construct an entity relation graph for each paragraph. The meta-path is defined as the path between two entities (e_i, e_j), where r_{ij} represents the relation between entities (appear in the same sentence).

$$e_i \xrightarrow{r_{i,i+1}} e_{i+1} \xrightarrow{r_{i+1,i+2}} \dots \xrightarrow{r_{j-1,j}} e_j$$

Based on the meta path, Jiao et al. constructed positive samples as context-option pairs, and generates negative samples by randomly replacing the sentences by a relation provider with the original entities. Finally, the positive samples and negative samples optimize a contrastive learning objective.

As previously mentioned, Jiao et al. have provided the pre-trained parameters of MERIt that can be easily fit into other fine-tuning models. In the original DAGN [5] approach, before the tokens are fed into the graph reasoning module, they are first fed into a large-scale pre-trained model. The original implementation utilize the RoBERTa [7] model for this step. Intuitively, the MERIt-enhanced RoBERTa model might provide more logical reasoning power to the DAGN approach. This modification is shown in Figure 2 (2).

V. Results

Since DAGN [5] has a more simple architecture and clear improving points, we base out experiments on the DAGN implementation published by the paper. We reproduced the models and try different improvement techniques and training strategies, getting results in Table II. We found that:

For hyper-parameter tuning,

1) Training Epochs (10 v.s. 15) for transformer downstream does not significantly affect the overall accuracy. (58.1% v.s. 58.0%).
2) Wider transformer fully-connected layer (1024 vs 2048) does not significantly affect the overall accuracy. (58.0% v.s. 57.7%).
3) Transformer performs differently from published GRU block in specific type of problems while maintaining a similar overall accuracy. (In Implication, Most Strongly Supported, Principle, and Match flaws tasks our model outperforms, while in Evaluation and Technique tasks our model under-performs, significantly, regarding original DAGN (acc difference $\tilde{1}$0%)). We interpret this difference as our Transformer block's focus on global attention and lack of focus on local tokens.
4) Enable pooling layer can improve the model performance slightly. The DAGN paper implements it by an extra dropout layer, but not enabled by default. We modified the model arguments to enable it, and got a more reasonable and closer reproduced result.

For different models,

1) Plain BERT baseline + dependency graph reaches accuracy 40.5%.

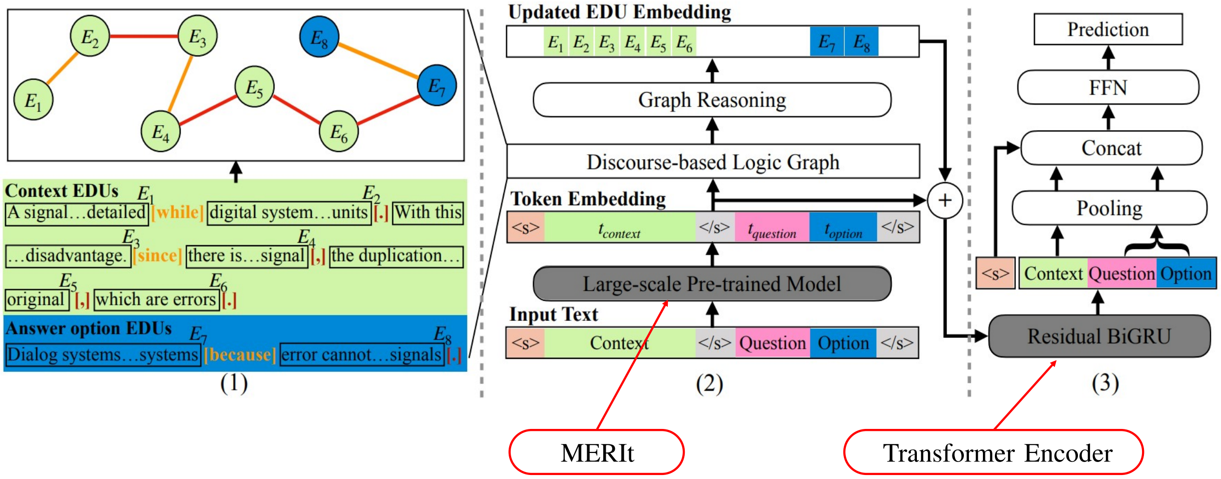

Fig. 2. Our modified DAGN model, utilizing MERIt pretrained model at the beginning and Transformer Encoder blocks in downstream model.

2) DAGN paper implementation reaches 58.2%, our reproduced version/plain transformer modification reaches 58.0%.
3) MERIt Pretrained + DAGN reaches $\tilde{5}$9.%.
4) MERIt Pretrained + DAGN transformer modification with pooling enabled reaches 60.3%, which is a considerable improvements upon official DAGN performance.

It is worth noting that, due to the large size of our model after introducing transformer blocks, the normal training process would run out of GPU Memory (14GiB, Tesla P100, on Google Colab Pro). The batch size in training, evaluation and testing phase is consequently decreased to 2 examples/batch, and our transformer uses 2 stacked encoder layers and 4 attention heads only.

In summary, MERIt Pretrained + DAGN transformer modification with pooling enabled reaches the best result in our experiments, achieving 2.1% higher accuracy than the DAGN's official performance.

TABLE II
EXPERIMENT RESULTS OF MODELS AND HYPER-PARAMETERS

Model Type	Test Acc.(%)
Baseline	40.5
DepReasoner	41.8
Official DAGN	58.2
DAGN+MERIt Pretrained	59.2
DAGN Reproduced (Without pooling)	57.7
DAGN Reproduced (With pooling)	58.0
DAGN+Transformer(10 epochs, w/o pooling)	58.1
DAGN+Transformer(15 epochs, w/o pooling)	58.0
DAGN+Transformer (10 epochs, 2048d FFN, w/o pooling)	57.7
DAGN+MERIt Pretrained+Transformer+Pooling	60.3

VI. FUTURE WORKS

Though achieving competitive results, there are still a few aspects that could be further investigated. Firstly, we notice that hierarchical learning could be introduced. For example, we could build a module that predicts the type of question. For human readers, the first step of solving the problem is to understand what the question is about, such as strengthening, weakening or comparing, etc. Currently, all of the models discussed above do not explicitly predict the type of question. It is also expected that, when explicitly integrating the question type to the input of neural architectures, the performance of the model would be further improved. Secondly, for EDU embedding, the positional embedding could also be aggregated, in the fashion of the positional encoding. Besides, our current training strategy might not be optimal. We notice that for the training process of our DepReasoner, the learning rate should be lowered to the scale of $1e-7$ in later epochs. However, we have not exploit these small learning rate on DAGN and MERIt, which remains to be a potential.

ACKNOWLEDGMENT

We thank Yu et al. for permission to use the ReClor dataset for our research. This project is supervised by Yik-Cheung Tam at NYU Shanghai.

REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017.

[2] P. Rajpurkar, R. Jia, and P. Liang, "Know what you don't know: Unanswerable questions for squad," *CoRR*, vol. abs/1806.03822, 2018.

[3] W. Yu, Z. Jiang, Y. Dong, and J. 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*, OpenReview.net, 2020.

[4] F. Jiao, Y. Guo, X. Song, and L. Nie, "Merit: Meta-path guided contrastive learning for logical reasoning," *CoRR*, vol. abs/2203.00357, 2022.

[5] Y. Huang, M. Fang, Y. Cao, L. Wang, and X. 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* (K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tür, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, and Y. Zhou, eds.), pp. 5848–5855, Association for Computational Linguistics, 2021.

[6] Y. Qin, Y. Lin, R. Takanobu, Z. Liu, P. Li, H. Ji, M. Huang, M. Sun, and J. Zhou, “Erica: Improving entity and relation understanding for pre-trained language models via contrastive learning,” May 2021.

[7] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “Roberta: A robustly optimized bert pretraining approach,” 2019. cite arxiv:1907.11692.

[8] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, “Albert: A lite bert for self-supervised learning of language representations.,” in *ICLR*, OpenReview.net, 2020.

[9] “Evaluating state of the art in ai.”

[10] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. v. d. Berg, I. Titov, and M. Welling, “Modeling relational data with graph convolutional networks,” 2017.

[11] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, “Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter,” *CoRR*, vol. abs/1910.01108, 2019.

[12] M. Honnibal and I. Montani, “spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.” To appear, 2017.

[13] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, “Gated graph sequence neural networks,” 2015.