Improving Verifiability of TRIP with Data Augmentation and Graph Neural Networks

Zhihao Xu
xuzhihao@umich.edu

Zheyu Zhang
zheyuz@umich.edu

Abstract

Large language models (LMs) have shown impressive performance on many language understanding tasks. However, the high accuracy achieved by LMs is evaluated on end tasks, which does not imply the model's ability in language understanding and reasoning. Tiered Reasoning for Intuitive Physics (TRIP) (Storks et al., 2021) is a commonsense reasoning dataset proposed to evaluate the multi-tiered performance of machines' reasoning processes. The empirical results on TRIP dataset demonstrate that LMs only achieve high-end task performance but struggle to provide supporting evidence. In this project, we experiment on TRIP dataset with two approaches, namely data augmentation and incorporating LMs and graph neural networks (GNNs), to improve the model's understanding and reasoning ability.

1 Introduction

Commonsense reasoning is a popular research topic in natural language understanding. Today's best models have already surpassed human performance in challenging language understanding tasks, including benchmarks for commonsense inference (Bowman et al., 2015; Zellers et al., 2018; Bhagavatula et al., 2019). However, there is still suspicion about whether these models have a deep understanding of the task (Bender and Koller, 2020; Linzen, 2020). It is still in doubt whether the problems are truly solved and whether these models can perform verifiable reasoning as humans do.

Tiered Reasoning for Intuitive Physics (TRIP) is a benchmark targeting physical commonsense reasoning, which proposes a high-level end task for story plausibility classification, a common proxy task for commonsense reasoning problems (Roemmele et al., 2011; Mostafazadeh et al., 2016; Sap et al., 2019b; Bisk et al., 2020a). The detailed task about TRIP is introduced in Section 2.1. The goal of TRIP is to help evaluate whether a high-level plausibility prediction can be verified based on lower-level understanding. The empirical results in (Storks et al., 2021) show that large LMs can achieve high end task performance (up to 78% accuracy), they struggle in the low-level understanding classification (only 11% accuracy in physical state classification). Hence, this model cannot be accounted to have a deep understanding of the task.

In order to improve the TRIP baseline model's commonsense understanding ability, we experimented on the TRIP dataset with two approaches, namely data augmentation techniques and incorporating LMs and GNNs for the physical state classification task.

2 Tiered Reasoning for Intuitive Physics

2.1 Dataset

The Tiered Reasoning for Intuitive Physics (TRIP) is a benchmark for physical commonsense reasoning. It proposes a high-level end task for story plausibility classification, a common proxy task for commonsense reasoning problems (Roemmele et al., 2011; Mostafazadeh et al., 2016; Sap et al., 2019b; Bisk et al., 2020a). The dataset contains highly similar story pairs, which describe a sequence of concrete physical actions. (differing only by one sentence which makes one of the stories implausible) An example is shown in Figure 1 (Storks et al., 2021), the Sentence 5 is different between two stories. In story B, Sentence 5 is conflicted with Sentence 2, which makes story B implausible. They hired several separate workers to each write a new sentence to replace a sentence in the original story and convert it to an implausible story. After conversion, the new story is no longer realistic in the physical world. The quality is ensured by a round of manual verification, all the typos are corrected and bad stories are removed.

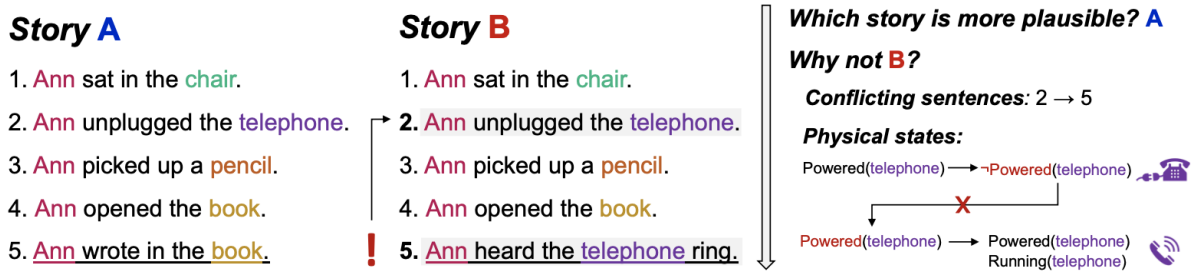


Figure 1: Story pair from TRIP, along with the tiers of annotation available to represent the reasoning process.

2.2 Problem statement

In this project, we mainly focus on three tasks proposed in (Storks et al., 2021).

Physical state classification. For each sentence-entity pair in each story, we need to do both precondition and effect state classification. For example, consider the entity *potato* in the sentence "John cut the cooked potato in half". We should predict the precondition is *solid* and the effect state is *in pieces*.

Conflict detection. Next we need to identify a pair of sentences in the form $S_i \rightarrow S_j$, where S_j is a *breakpoint* and S_i is the *evidence* that explains why the *breakpoint* is a conflict. For example, in Figure 1, Sentence 5 is a *breakpoint* and Sentence 2 is the *evidence*. In the TRIP dataset, one conflicting sentence pair is sufficient for conflict.

Story classification The final task is to classify which story is the story pair is plausible. This can be achieved based on the number of conflicts detected in each two stories.

The baseline model proposed in (Storks et al., 2021) tried to solve the three tasks above, but still have three major limitations: (1) the accuracy of physical state classification (verifiability) is poor, (2) overfitting exists in conflict detection, (3) the consistency and verifiability on low-level task cannot be transmitted to the high-level task. In this project, we mainly focus on the first two limitations and propose some approaches to resolve them.

3 Related Work

Physical commonsense. There are several existing physical commonsense reasoning datasets focusing on various classification tasks. Propara (Mishra et al., 2018) introduces a text in which existence and location of entities are tracked in each sentence. Physical Interaction Question Answering (PIQA) (Bisk et al., 2020b) develops a physical commonsense high-level end task of multiple-choice text plausibility classification. Other datasets focus on specific domains of physical commonsense, such as temporal reasoning (Zhou et al., 2019), spatial reasoning (Mirzaee et al., 2021), visual reasoning (Johnson et al., 2017; Bakhtin et al., 2019), and multi-modal reasoning (Hudson and Manning, 2019; Das et al., 2018; Anderson et al., 2018; Shridhar et al., 2020).

Robust language inference. Several works focus on examine the robustness of contextual language embeddings through syntactic and semantic phenomena (Adi et al., 2016; Ettinger et al., 2018; Tenney et al., 2019b; Hewitt and Manning, 2019; Jawahar et al., 2019; Tenney et al., 2019a), and others have explored specialized natural language inference tasks (Welleck et al., 2018; Uppal et al., 2020) and logic rules (Li et al., 2019; Asai and Hajishirzi, 2020) for evaluating robustness and consistency. Some approaches are proposed to improve model robustness against exploiting various types of biases (Belinkov et al., 2019; Clark et al., 2019; Min et al., 2020). Recent works focus on providing knowledge-supprted language understanding by compiling large amounts of semi-structured commonsense knowledge (Sap et al., 2019a; Mostafazadeh et al., 2020) and injecting this knowledge into pre-trained language models (Bosselut et al., 2019; Zhang et al., 2019).

Knowledge-aware methods for natural language processing Several existing approaches study pretrained LMs' potential as latent knowledge bases (Pan et al., 2019; Ye et al., 2019; Petroni et al., 2019; Bosselut et al., 2019), and other works integrate knowledge graphs into LMs (Mihaylov and Frank, 2018; Lin et al., 2019; Wang et al., 2019; Yang et al., 2019; Wang et al., 2020b; Bosselut et al., 2021).

Graph neural networks Several studies leverage GNNs to model the structure of text or knowledge graphs (Yasunaga et al., 2017; Zhang et al., 2018; Yasunaga and Liang, 2020; Wang et al., 2020a).

4 Approaches

4.1 Data Augmentation

In the conflict detection problem, the validation loss increase as the iteration increases (Storks et al., 2021), which indicates that the model is over-fitting on the training data. The cost of collecting more story pairs is relatively expensive, hence, we plan to use data augmentation techniques to resolve this. In this project, we first plan to use some simple data augmentation such as back translation, synonym replacement, etc. Furthermore, we also plan to apply some of the ideas used in CV data augmentation, such as sentence shuffling, word swapping, etc.

Easy Data Augmentation (EDA) Easy Data Augmentation Techniques are proposed to improve the performance on text classification task. It consists of four simple operations (Wei and Zou, 2019).

1. **Synonym Replacement (SR):** Randomly choose n words from the sentence that are not stop words. Replace each of these words with one of its synonyms chosen at random.

2. **Random Insertion (RI):** Find a random synonym of a random word in the sentence that is not a stop word. Insert that synonym into a random position in the sentence. Do this n times.

3. **Random Swap (RS)**: Randomly choose two words in the sentence and swap their positions. Do this n times.

4. **Random Deletion (RD):** Randomly remove each word in the sentence with probability p.

In order to keep the balance between the long sentences and short sentences, we vary the number of words changed, n, for SR, RI, and RS based on the sentence length l with the formula $n = pl$, where p is the parameter that indicates the percent of the words in a sentence that changed. Furthermore, for each original sentence, we generate n_{aug} augmented sentences.

Back Translation Back translation is another augmentation technique that can be used to enrich the dataset and boost the classification performance (Beddiar et al., 2021). We first translate the original sentence to a second language then translate back to English.

4.2 Incorporating Large Language Models (LMs) and Graph Neural Networks (GNNs)

In the physical state classification (i.e., precondition and effect state classification), the TRIP baseline models treat each (entity, sentence) pair independently and input the contextual embedding to a two-layer feed-forward neural network. However, it is common that an entity's physical states are affected by its relation with other entities in the sentences. For example, the physical state of the entity *telephone* in the sentence "*Ann unplugged the telephone.*" is intuitively correlates with the verb *unplugged*. Therefore, our goal is to extract such useful relations between entities in a sentence and inference the physical states of entities. Recently, Graph Neural Networks (GNNs) have demonstrated the power of extracting the relational information in a graph, and studies in the question answering (QA) field (Lin et al., 2019; Feng et al., 2020; Yasunaga et al., 2021) have indicated that deep learning models incorporating both LMs and GNNs have a stronger commonsense question answering ability, as LMs encode unstructured knowledge implicitly whereas GNNs extract structured knowledge explicitly presented in a graph (e.g., knowledge graph). Inspired by the aforementioned works, we will try to improve the verifiability of the baseline model's physical state classifier by incorporating both LMs and GNNs. Based on the idea, our method contains two main steps, namely constructing useful graphs based on sentences and designing a physical classification model incorporating LMs and GNNs.

4.2.1 Graph Construction

In order to leverage GNNs to extract relational information between entities in a sentence, we need to first construct an informative graph that encodes the entity relations in the sentence. In the question-answering (QA) task, deep learning models incorporating LMs and GNNs (Lin et al., 2019; Feng et al., 2020; Yasunaga et al., 2021) utilize the ConceptNet (Speer et al., 2017), a general domain knowledge graph as the external knowledge

graph (KGs) to construct the graph of a QA context. Specifically, they first recognize entities mentioned in the QA context and initialize the node set $\mathcal{V}$ by linking them to the entities in ConceptNet. For each node in $\mathcal{V}$, its neighboring nodes within two hops in ConceptNet are also added to $\mathcal{V}$. The graph $\mathcal{G}$ of the QA context is constructed by extracting a subgraph from ConceptNet based on node set $\mathcal{V}$.

However, the aforementioned graph construction strategy cannot be directly applied to our setting, as our goal of graph construction is to encode entity relations in a sentence, whereas the goal in the QA task is to encode the relational path information between entities in the question and answer context. As the dependency structure of a sentence s reflects the word relations in the sentence, we first perform dependency parsing on the sentence and extract the dependency graph $\mathcal{G}_s = \{\mathcal{V}_s, \mathcal{E}_s\}$, with nodes $v \in \mathcal{V}_s$ representing words and directed edges $e \in \mathcal{E}_s$ representing dependencies. To further incorporate commonsense knowledge, we assign each node $v \in \mathcal{V}_s$ with its ConceptNet Numberbatch embedding x_v, the word embeddings pretrained on ConceptNet, as its attributes. Therefore, the constructed graph (Figure 2) has both word relations from the dependency graph structure and commonsense knowledge from node attributes generated from ConceptNet.

4.2.2 Model Architecture for Physical Classification

Based on the contextual embedding obtained from LMs constructed graph with node attributes, the modified physical classification model contains two main parts: a k-layer GNN model for extracting relational information from the constructed graph, and a deep neural network model to predict an entity's physical states based on its contextual embedding and latent embedding from GNN model. The model architecture with an input example is shown in Figure 2, and the detailed algorithm is displayed in Algorithm 1.

Graph Neural Network Model The general graph neural network model uses a form of *neural message passing* (Gilmer et al., 2017), where nodes aggregate features from their neighboring nodes and update the latent representations using a nonlinear activation function. During each message-passing iteration of the GNN model, the latent representation of a node is updated according to information from aggregated from its neighboring

Algorithm 1 Modified physical state classifier

1: **Input:** Entity v, Entity v, dependency graph $\mathcal{G}_s$, v's attribute $\mathbf{x}_v$
2: **Output:** Predicted precondition state $\mathbf{y}_v^{pre}$, effect state $\mathbf{y}_v^{eff}$ of entity v.
3: **for** $k = 0, \ldots, K-1$ **do**
4: Calculate weight $\alpha_{v,u}, u \in \mathcal{N}(v)$ (Eq. 4)
5: Compute message $\mathbf{m}_{\mathcal{N}(v)}^k$ (Eq. 3)
6: Obtain $\mathbf{h}_v^{(k+1)}$ (Eq. 2)
7: $\mathbf{z}_v = \mathbf{h}_v^{(K)}$
8: $\mathbf{c}_v = \mathbf{t}_v \oplus \mathbf{z}_v$
9: Compute $\mathbf{y}_v^{pre}$, $\mathbf{y}_v^{eff}$ (Eq. 5)

nodes. In particular, given the dependency graph $\mathcal{G}_s = \{\mathcal{V}_s, \mathcal{E}_s\}$ of a sentence s, along with the node attribute matrix $\mathbf{X}_s \in R^{|\mathcal{V}_s| \times d}$, a GNN layer is represented as (Hamilton, 2020)

$$\mathrm{h}_v^{(k+1)} = \mathrm{UPDATE}^{(k)}(\mathbf{h}_v^{(k)}, \mathrm{AGG}^{(k)}(\{\mathbf{h}_u^{(k)}, \forall u \in \mathcal{N}(v)\})), \tag{1}$$

where $\mathbf{h}_v^{(k)}$ denotes the latent representation of node v in the k-th layer ($\mathbf{h}_v^{(0)} = \mathbf{x}_v$), and $\mathcal{N}(v)$ represents the set of neighboring nodes of node v. UPDATE and AGG denotes the update and aggregate functions, repspectively, and they can be any differentiable function (e.g., ReLU for UPDATE and sum/mean for AGG). If we further let the message function be $\mathbf{m}_{\mathcal{N}(v)}^{(k)} = \mathrm{AGG}^{(k)}(\{\mathbf{h}_u^{(k)}, \forall u \in \mathcal{N}(v)\})$, the formula of a GNN layer becomes

$$\mathbf{h}_v^{(k+1)} = \mathrm{UPDATE}^{(k)}(\mathbf{h}_v^{(k)}, \mathbf{m}_{\mathcal{N}(v)}^{(k)}). \tag{2}$$

The final representation of node v after running a K-layer GNN model is $\mathbf{z}_v = \mathbf{h}_v^{(K)}$.

In our method, Graph Attention Networks (GAT) is selected as the GNN model to learn the entity relations in the constructed dependency graph. In particular, the message function of the GAT model is a weighted sum function over the neighboring nodes' features, denoted as

$$\mathbf{m}_{\mathcal{N}(v)} = \sum_{u \in \mathcal{N}(v)} \alpha_{v,u} \mathbf{h}_v, \tag{3}$$

where $\alpha_{v,u}$ denotes the attention weight, and it is computed as

$$\alpha_{v,u} = \frac{\exp\big(\mathrm{LeakyReLU}\big(\mathbf{a}^\top [\mathbf{W}\mathbf{x}_v \oplus \mathbf{W}\mathbf{x}_u]\big)\big)}{\sum_{k \in \mathcal{N}(v) \cup \{v\}} \exp\big(\mathrm{LeakyReLU}\big(\mathbf{a}^\top [\mathbf{W}\mathbf{x}_v \oplus \mathbf{W}\mathbf{x}_k]\big)\big)}, \tag{4}$$

where $\mathbf{a}$ is a trainable attention vector, $\mathbf{W}$ represents the trainable weight matrix, and $\oplus$ denotes the

concatenation operation. Compared to Graph Convolutional Networks (GCNs) which assign equal weights to neighboring nodes in the aggregation process, GAT learns the attention weights based on the similarity between nodes and is more appropriate in our setting as we want to pay more attention to the words having higher similarity with the target word in the neighborhood aggregation process.

Deep Neural Network Model The Deep Neural Network Model is similar to the original physical state classifier presented in the TRIP baseline model, and the only difference is the input feature vector. In our model, we first concatenate the contextual embedding with the entity representation obtained from the GNN model and then feed the concatenated embedding to precondition and effect classifiers, which are typical feed-forward classification heads with one precondition classifier and one effect classifier for each of the 20 physical attribute tracked in the TRIP dataset. In particular, for an entity v, the output representation of GNN model is $\mathbf{z}_v = \mathbf{h}_v^{(K)}$, and its contextual embedding is $\mathbf{t}_v$. The concatenated embedding $\mathbf{c}_v = \mathbf{z}_v \oplus \mathbf{t}_v$. Next, we feed $\mathbf{c}_v$ to precondition and effect classifiers to predict the precondition and effect state of entity v,

$$\mathbf{y}_v^{pre} = \mathcal{F}_{pre}(\mathbf{c}_v) + \mathbf{b}_{pre}, \tag{5a}$$

$$\mathbf{y}_v^{eff} = \mathcal{F}_{eff}(\mathbf{c}_v) + \mathbf{b}_{eff}. \tag{5b}$$

5 Evaluation

In this section, we will evaluate the performance (i.e., verifiability, consistency, accuracy) on the TRIP dataset after performing data augmentation (Section 5.1) and modifying the physical state classifier with GNN models (Section 5.2).

5.1 Evaluation on Data Augmentation

In this experiment, we use both EDA and back-translation to enrich the training dataset. In EDA, we choose n_{aug}, which means we expand the training data four times larger. In order to avoid making the plausible story implausible, we only do the augmentation on the implausible stories. As for the four simple operations, we choose a high proportion ($p = 0.4$) for the Synonym Replacement and a low proportion ($p = 0.05$) for the rest three operation. That's because the other three augmentation operations might also influence the physical state, which is difficult to check the correctness. Moreover, we also applied the back translation once on the training data. Totally, the training data size is increased to five times the original training data size.

Figure 3 shows the comparison between the loss of the original TRIP model and the augmented TRIP model. From the sub-figure (A), we can see the decrease of physical states loss. This indicates that with more training data and more iterations, we can achieve lower physical state loss. However, one of the main limitations of the original TRIP model is the over-fitting in conflict detection loss. Unfortunately, data augmentation did not resolve this issue quite well. The over-fitting issue still exists in the conflict detection task. From the sub-figure (B), we can see that both the original TRIP experiment and the data augmentation experiment can achieve minimal loss after around 4000 iterations, and the validation loss increases after that. The result in sub-figure (C) is also very similar. The training loss decreased in the story classification task but the validation did not. These might be caused by the similarity between the original training data and the augmentation data. If the generated sentence is too similar to the original one, the performance of the model will be similar to directly going through the same training data multiple times. We also trained to increase the ratio p in EDA but the performance is still similar to the original one. Hence, we think that EDA and back translation might not be a great approach to resolve the over-fitting issue in this task. Further work may consider more complicated data augmentation techniques or collecting more training data to resolve this.

5.2 Evaluation on Modified Physical State Classifier

Data Preprocessing First, we need to construct the dependency graphs of sentences with ConceptNet Numberbatch attributes according to Section 4.2.1. The Stanford Dependency Parser (Manning et al., 2014) API in the NLTK package (Bird et al., 2009) is utilized to perform dependency parsing and extract dependency graphs for sentences. The gensim package (Řehůřek and Sojka, 2010) is used to extract the ConceptNet Numberbatch embeddings (Speer et al., 2017) for each word in the sentences. Regarding words not included in the Con-

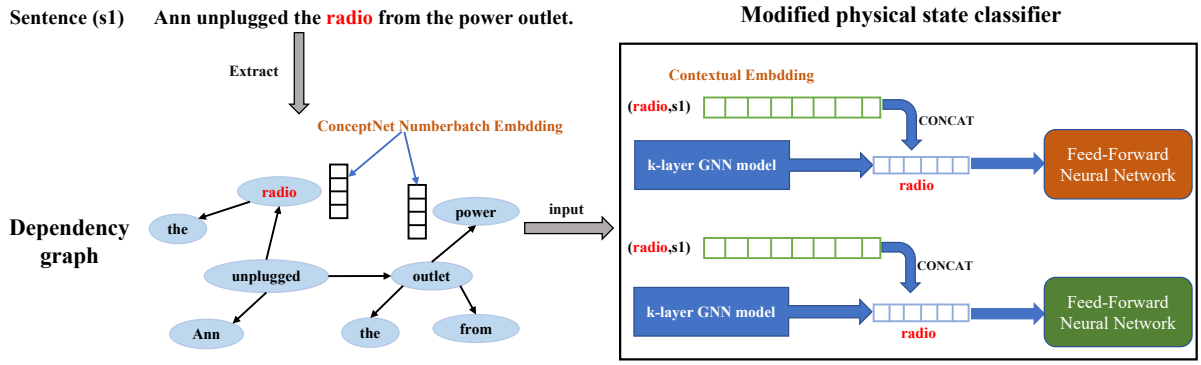


Figure 2: The procedure of incorporating GNNs and LMs in the physical state classification, with an example of predicting the precondition and effect state of entity "radio" in the sentence "Ann unplugged the radio from the power outlet.".

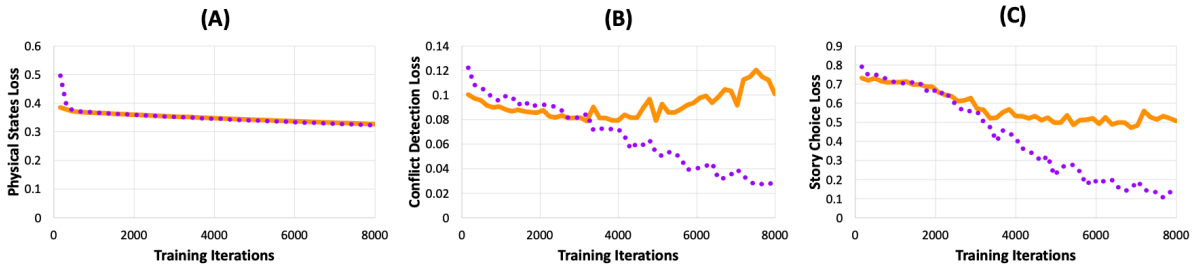
ceptNet Numberbatch embedding file, we transform these words to their original form and extract embedding (e.g., "neighbor's" to "neighbor"). For each sentence in a story, we generate its corresponding dependency graph with ConceptNet Numberbatch features.

Experimental Setup Our implementation is based on the TRIP baseline model, where we modified the transformer backbone architecture, featurize function, data loader, and evaluation functions, etc. to incorporate the GNN model into the physical state classifier. Regarding the GNN model, we choose a 2-layer GAT model with 8 heads. The main challenge when implementing the modified physical state classifier is that the entities of stories in the TRIP dataset can be either a word or phrases. However, the nodes in the dependency graph of a sentence represent words. For example, "washing machine" is an entity in the TRIP dataset, whereas the dependency graph only contains nodes "washing" and "machine". Therefore, we cannot directly obtain the embedding of phrase entities in the dataset. To resolve this problem, we propose two solutions. The first solution is relatively simple. Given a phrase as an entity in the dataset, if all of the words in the phrase exist in the graph, we average the latent representation of the words in the phrase as the latent representation of the phrase. Otherwise, we use a feed-forward neural network instead, with input features as the ConceptNet Numberbatch word embedding. The second solution maximizes the use of the GNN model and takes a longer time for training. Specifically, given a phrase, we feed all the words in the phrases to the GNN model if they exist in the dependency graph, and feed the other nodes to a feed-forward neural network. Latent representations of all the words in the phrase are averaged and serve as the phrase's latent representation. We denote the first solution as "simple", and the second solution as "complex". In this experiment, we will implement RoBERTa(simple), DeBERTa(simple), and RoBERTa(complex) these three models.

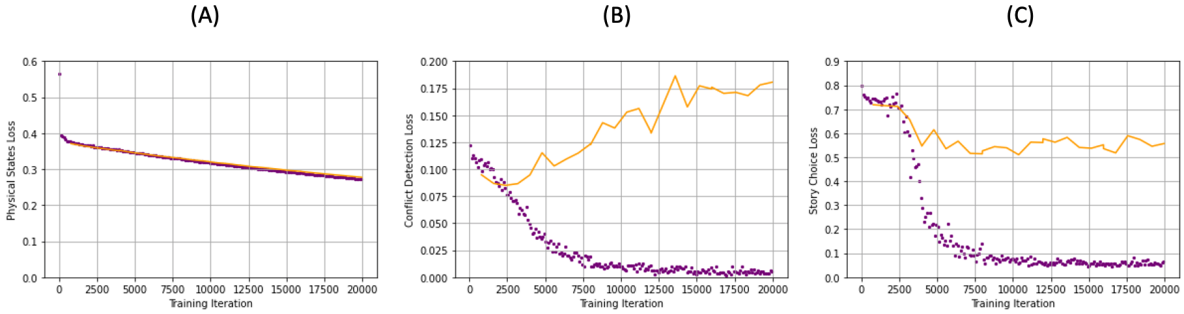
Results The evaluation results are shown in Table 1. It can be seen that our implemented models with modified physical state classifier do not demonstrate the superior performance compared to the baseline model, and even downgrade the verifiability for our RoBERTa(simple) and RoBERTa(complex) models. Taking a further look at the training and validation loss on the physical state classification of the RoBERTa(simple) model (Figure 4), the loss increases compared to the TRIP baseline model, which indicates that the current modification of the physical classification model actually introduces noise to the TRIP data and downgrades the performance on the physical state classification task. However, regarding the accuracy and consistency, it can be seen that incorporating the GNN models is able to increase the model performance.

Interpretation of experimental results Based on the evaluation results, we have the following thoughts and interpretations:

- The current dependency graph construction technique introduces noise to the TRIP data. As mentioned in **Experimental Setup** paragraph, the current dependency parsing technique only extracts words and cannot extract phrases, and ConceptNet Numberbatch also



(a) Experiment Result from Figure 4 in TRIP paper (Storks et al., 2021)



(b) Experiment Result of Data Augmentation

Figure 3: Training (purple, dotted) and validation (orange, solid) losses for best tiered ROBERTA system trained on TRIP for 10 epochs. (A) physical state classification, (B) conflict detection, and (C) story choice classification.

Model	**Accuracy** (%)	**Consistency** (%)	**Verifiability** (%)
RoBERTa(baseline)	75.2	18.8	5.7
RoBERTa(simple)	73.9	22.0	4.34
DeBERTa(simple)	67.8	14.8	1.14
RoBERTa(complex)	76.1	21.6	3.42

Table 1: Evaluation metrics for tiered models on the test set of TRIP. Compared to the RoBERTa baseline models.

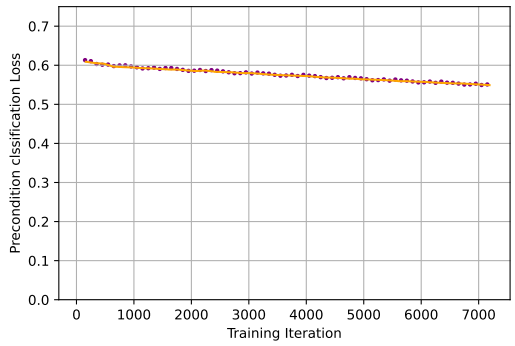
does not have pre-trained embeddings for phrases. Therefore, such extracted dependency graph introduces noisy information regarding the phrase entities in the TRIP dataset. In addition, the Stanford dependency parser tends to make small mistakes when the input sentence has a complex structure, which results in noisy relational information on the extracted dependency graph.

- Precondition and effect state classifiers on the same graph hinder the improvement of the verifiability of the model. Currently, we input the same dependency graph to precondition and effect state classifiers. As these two classifiers share the same initial relational information, leveraging GNN models can be regarded as a similarity augmentation, which prevents these two classifiers from distinguishing the precondition and effect states of entities.

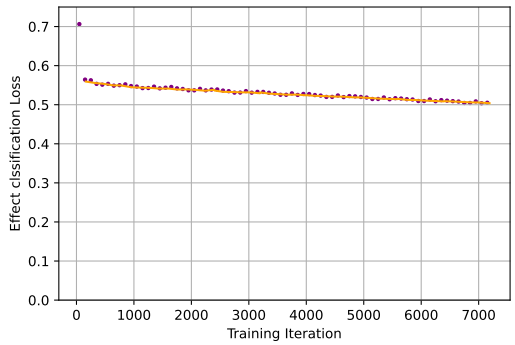
6 Conclusion and Discussion

In this project, we implemented two methods, namely data augmentation and incorporating GNNs into physical state classifier, to the TRIP dataset to resolve the over-fitting issue on the conflict detection and improve the verifiability. Experiments demonstrate that neither data augmentation nor incorporating GNNs into the baseline model improve its verifiability on the TRIP dataset. However, experimental results still provide useful information on improving the model performance on the TRIP dataset for future study. Regarding the data augmentation experiments, the result indicates that augmenting the TRIP dataset with similar sentences cannot resolve the over-fitting issue on conflict detection. Instead, we should collect more training data or generate augmented sentences with lower similarities in further study.

Regarding the idea of incorporating LMs and



(a) Precondition state classification



(b) Effect state classification

Figure 4: Training (purple, dotted) and validation (orange, solid) losses on (a) precondition classification and (b) effect state classification for best ROBERTA(simple) model trained on TRIP for 10 epochs.

GNNs, the current designed models do not perform well on the physical state classification task. Nevertheless, with interpretation and understanding of results discussed in Section 5.2, we may improve the model verifiability from the following aspects,

- Develop more accurate approaches to extract/construct informative graphs from sentences. Relation types may be extracted from knowledge graphs (KG) to enhance the true relation in the dependency graph, and we may build a more useful graph by leveraging the external KG resources. For example, we may develop a KG subgraph extraction algorithm specially for the TRIP data.

- Utilize the relational graph neural network (Schlichtkrull et al., 2018) or KG embeddings (Wang et al., 2017) as node attributes.

- Incorporate external resources to construct different graphs of a given sentence for precondition and effect state classifications, respectively.

- Design an algorithm for dynamically updating the graph based on sentences to reduce the model's computational complexity. The current designed model takes a large amount of time for training, as each sentence in a story corresponds to a graph.

7 Division of Work

Work division

Data preprocessing and data augmentation techniques are implemented by Zhihao Xu. The dependency graph construction and incorporating LMs and GNNs into physical state classification are implemented by Zheyu Zhang. Our source code public available at https://github.com/HowIII/EECS595_project_group28. As for the final report, in the Approaches and Evaluation Section, data augmentation related sub-section (Section 4.1 and 5.1) are written by Zhihao Xu, graph neural network related sub-section (Section 4.2 and 5.2) are written by Zheyu Zhang. All the other sections are written jointly.

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2016. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *arXiv preprint arXiv:1608.04207*.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. 2018. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3674–3683.

Akari Asai and Hannaneh Hajishirzi. 2020. Logic-guided data augmentation and regularization for consistent question answering. *arXiv preprint arXiv:2004.10157*.

Anton Bakhtin, Laurens van der Maaten, Justin Johnson, Laura Gustafson, and Ross Girshick. 2019. Phyre: A new benchmark for physical reasoning. *Advances in Neural Information Processing Systems*, 32:5082–5093.

Djamila Romaissa Beddiar, Md Saroar Jahan, and Mourad Oussalah. 2021. Data expansion using back translation and paraphrasing for hate speech detection. *Online Social Networks and Media*, 24:100153.

Yonatan Belinkov, Adam Poliak, Stuart M Shieber, Benjamin Van Durme, and Alexander M Rush. 2019. Don't take the premise for granted: Mitigating artifacts in natural language inference. *arXiv preprint arXiv:1907.04380*.

Emily M Bender and Alexander Koller. 2020. Climbing towards nlu: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Scott Wen-tau Yih, and Yejin Choi. 2019. Abductive commonsense reasoning. *arXiv preprint arXiv:1908.05739*.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Yonatan Bisk, Ari Holtzman, Jesse Thomason, Jacob Andreas, Yoshua Bengio, Joyce Chai, Mirella Lapata, Angeliki Lazaridou, Jonathan May, Aleksandr Nisnevich, et al. 2020a. Experience grounds language. *arXiv preprint arXiv:2004.10151*.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020b. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7432–7439.

Antoine Bosselut, Ronan Le Bras, and Yejin Choi. 2021. Dynamic neuro-symbolic knowledge graph construction for zero-shot commonsense question answering. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. Comet: Commonsense transformers for automatic knowledge graph construction. *arXiv preprint arXiv:1906.05317*.

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*.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2019. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. *arXiv preprint arXiv:1909.03683*.

Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. 2018. Embodied question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–10.

Allyson Ettinger, Ahmed Elgohary, Colin Phillips, and Philip Resnik. 2018. Assessing composition in sentence vector representations. *arXiv preprint arXiv:1809.03992*.

Yanlin Feng, Xinyue Chen, Bill Yuchen Lin, Peifeng Wang, Jun Yan, and Xiang Ren. 2020. Scalable multi-hop relational reasoning for knowledge-aware question answering. *arXiv preprint arXiv:2005.00646*.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR.

William L Hamilton. 2020. Graph representation learning. *Synthesis Lectures on Artifical Intelligence and Machine Learning*, 14(3):1–159.

John Hewitt and Christopher D Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138.

Drew A Hudson and Christopher D Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does bert learn about the structure of language? In *ACL 2019-57th Annual Meeting of the Association for Computational Linguistics*.

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. 2017. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2901–2910.

Tao Li, Vivek Gupta, Maitrey Mehta, and Vivek Srikumar. 2019. A logic-driven framework for consistency of neural models. *arXiv preprint arXiv:1909.00126*.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. Kagnet: Knowledge-aware graph networks for commonsense reasoning. *arXiv preprint arXiv:1909.02151*.

Tal Linzen. 2020. How can we accelerate progress towards human-like linguistic generalization? *arXiv preprint arXiv:2005.00955*.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

Todor Mihaylov and Anette Frank. 2018. Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge. *arXiv preprint arXiv:1805.07858*.

Junghyun Min, R Thomas McCoy, Dipanjan Das, Emily Pitler, and Tal Linzen. 2020. Syntactic data augmentation increases robustness to inference heuristics. *arXiv preprint arXiv:2004.11999*.

Roshanak Mirzaee, Hossein Rajaby Faghihi, Qiang Ning, and Parisa Kordjmashidi. 2021. Spartqa:: A textual question answering benchmark for spatial reasoning. *arXiv preprint arXiv:2104.05832*.

Bhavana Dalvi Mishra, Lifu Huang, Niket Tandon, Wen-tau Yih, and Peter Clark. 2018. Tracking state changes in procedural text: a challenge dataset and models for process paragraph comprehension. *arXiv preprint arXiv:1805.06975*.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and evaluation framework for deeper understanding of commonsense stories. *arXiv preprint arXiv:1604.01696*.

Nasrin Mostafazadeh, Aditya Kalyanpur, Lori Moon, David Buchanan, Lauren Berkowitz, Or Biran, and Jennifer Chu-Carroll. 2020. Glucose: Generalized and contextualized story explanations. *arXiv preprint arXiv:2009.07758*.

Xiaoman Pan, Kai Sun, Dian Yu, Jianshu Chen, Heng Ji, Claire Cardie, and Dong Yu. 2019. Improving question answering with external knowledge. *arXiv preprint arXiv:1902.00993*.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI Spring Symposium Series*.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2019a. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3027–3035.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. 2019b. Socialiqa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. 2020. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10740–10749.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-first AAAI conference on artificial intelligence*.

Shane Storks, Qiaozi Gao, Yichi Zhang, and Joyce Chai. 2021. Tiered reasoning for intuitive physics: Toward verifiable commonsense language understanding. *arXiv preprint arXiv:2109.04947*.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019a. Bert rediscovers the classical nlp pipeline. *arXiv preprint arXiv:1905.05950*.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R Bowman, Dipanjan Das, et al. 2019b. What do you learn from context? probing for sentence structure in contextualized word representations. *arXiv preprint arXiv:1905.06316*.

Shagun Uppal, Vivek Gupta, Avinash Swaminathan, Haimin Zhang, Debanjan Mahata, Rakesh Gosangi, Rajiv Shah, and Amanda Stent. 2020. Two-step classification using recasted data for low resource settings. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 706–719.

Hongwei Wang, Hongyu Ren, and Jure Leskovec. 2020a. Entity context and relational paths for knowledge graph completion. *arXiv preprint arXiv:2002.06757*.

Peifeng Wang, Nanyun Peng, Filip Ilievski, Pedro Szekely, and Xiang Ren. 2020b. Connecting the dots: A knowledgeable path generator for commonsense question answering. *arXiv preprint arXiv:2005.00691*.

Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743.

Xiaoyan Wang, Pavan Kapanipathi, Ryan Musa, Mo Yu, Kartik Talamadupula, Ibrahim Abdelaziz, Maria Chang, Achille Fokoue, Bassem Makni, Nicholas Mattei, et al. 2019. Improving natural language inference using external knowledge in the science questions domain. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7208–7215.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*.

Sean Welleck, Jason Weston, Arthur Szlam, and Kyunghyun Cho. 2018. Dialogue natural language inference. *arXiv preprint arXiv:1811.00671*.

An Yang, Quan Wang, Jing Liu, Kai Liu, Yajuan Lyu, Hua Wu, Qiaoqiao She, and Sujian Li. 2019. Enhancing pre-trained language representations with rich knowledge for machine reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2346–2357.

Michihiro Yasunaga and Percy Liang. 2020. Graph-based, self-supervised program repair from diagnostic feedback. In *International Conference on Machine Learning*, pages 10799–10808. PMLR.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. Qa-gnn: Reasoning with language models and knowledge graphs for question answering. *arXiv preprint arXiv:2104.06378*.

Michihiro Yasunaga, Rui Zhang, Kshitijh Meelu, Ayush Pareek, Krishnan Srinivasan, and Dragomir Radev. 2017. Graph-based neural multi-document summarization. *arXiv preprint arXiv:1706.06681*.

Zhi-Xiu Ye, Qian Chen, Wen Wang, and Zhen-Hua Ling. 2019. Align, mask and select: A simple method for incorporating commonsense knowledge into language representation models. *arXiv preprint arXiv:1908.06725*.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. Swag: A large-scale adversarial dataset for grounded commonsense inference. *arXiv preprint arXiv:1808.05326*.

Yuhao Zhang, Peng Qi, and Christopher D Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. *arXiv preprint arXiv:1809.10185*.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. Ernie: Enhanced language representation with informative entities. *arXiv preprint arXiv:1905.07129*.

Ben Zhou, Daniel Khashabi, Qiang Ning, and Dan Roth. 2019. " going on a vacation" takes longer than" going for a walk": A study of temporal commonsense understanding. *arXiv preprint arXiv:1909.03065*.