Compositional Generalization for Kinship Prediction through Data Augmentation

Kangda Wei Sayan Ghosh Shashank Srivastava
UNC Chapel Hill
kangda@live.unc.edu, {sayghosh, ssrivastava}@cs.unc.edu

Abstract

Transformer-based models have shown promising performance in numerous NLP tasks. However, recent work has shown the limitation of such models in showing compositional generalization, which requires models to generalize to novel compositions of known concepts. In this work, we explore two strategies for compositional generalization on the task of kinship prediction from stories: (1) data augmentation and (2) predicting and using intermediate structured representation (in form of kinship graphs). Our experiments show that data augmentation boosts generalization performance by around 20% on average relative to a baseline model from prior work not using these strategies. However, predicting and using intermediate kinship graphs leads to a deterioration in the generalization of kinship prediction by around 50% on average relative to models that only leverage data augmentation.

1 Introduction

Transformer-based large language models (Vaswani et al., 2017) have achieved state-of-the-art results on numerous NLP tasks such as question answering, reading comprehension, relational reasoning, etc. that require both syntactic and semantic understanding of language. However, recent works (Bahdanau et al., 2018; Lake and Baroni, 2018; Gururangan et al., 2018; Kaushik and Lipton, 2018) have shown that these transformer-based models have their limitations when it comes to tasks that require compositional generalization as they often perform surface-level reasoning instead of understanding the underlying concepts and learning to generalize and reason over them. On the other hand, neural models that encode the structure of the data (such as Graph Attention Networks (Veličković et al., 2017)) instead of consuming it in an unstructured format

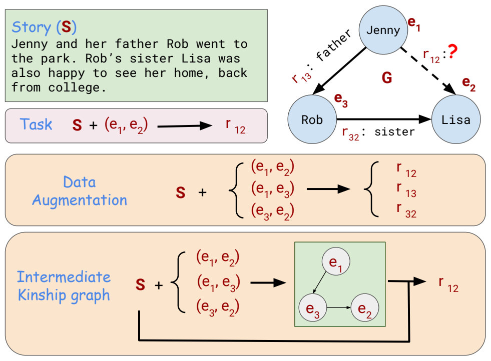

Story (S)
Jenny and her father Rob went to the park. Rob's sister Lisa was also happy to see her home, back from college.
Task
$S + (e_1, e_2) \longrightarrow r_{12}$
Jenny
e_1
r_{13}: father
r_{12}: ?
G
e_2
e_3
Rob
Lisa
r_{32}: sister
Data Augmentation
$S + \{(e_1, e_2), (e_1, e_3), (e_3, e_2)\} \longrightarrow \{r_{12}, r_{13}, r_{32}\}$
Intermediate Kinship graph
$S + \{(e_1, e_2), (e_1, e_3), (e_3, e_2)\} \longrightarrow$
e_1
e_3
e_2
r_{12}

Figure 1: To improve the compositional generalization of models for the task of kinship prediction between a pair of queried entities (e.g. predicting the relation r_{12} given the entities e_1 and e_2) from a story (S) we present two strategies (1) data augmentation and (2) predicting and using intermediate structured representation in form of kinship graphs. For data augmentation (first strategy), we utilize the existing ground truth graph (G) to generate more pairs of target relations and query entities (such as predicting r_{13} using e_1 and e_3) that do not need compositional inference to obtain the answer. In our second strategy, using our augmented data we predict an intermediate kinship graph and reason over it jointly with the story to predict the relation between the queried pair of entities.

show better compositional generalization (Sinha et al., 2019).

In this work, we explore two strategies to improve the compositional generalization of models for the task of kinship prediction from stories. In our first strategy, we explore the utility of data augmentation towards compositional generalization. Recent works have shown data augmentation to be an effective strategy in improving model performance on different NLP tasks such as Neural Machine Translation (Fernando and Ranathunga, 2022), semantic parsing (Yang et al., 2022), and text summarization (Wan and Bansal, 2022). Our data augmentation strategy focuses on improving a model's ability to extract relations that are explicitly mentioned in the text. In our second strategy, we explore the utility of predicting an intermediate structured representation of the story (as a kinship graph) and then jointly reasoning over it along with the story text for the task of kinship prediction. Figure 1 provides an example of this task and also illustrates the two strategies. The strategies are explained in detail in §3.

We evaluate the utility of our strategies on a kinship prediction benchmark, CLUTRR (Sinha et al., 2019). Overall, we find data augmentation is helpful and boosts the generalization performance (accuracy of predicting correct relation) by around 20% on average relative to a baseline not using these strategies. However, using intermediate kinship graphs deteriorates generalization performance by almost 50% as compared to the model that only uses data augmentation. Our code is available at: `https://github.com/WeiKangda/data-aug-clutrr`.

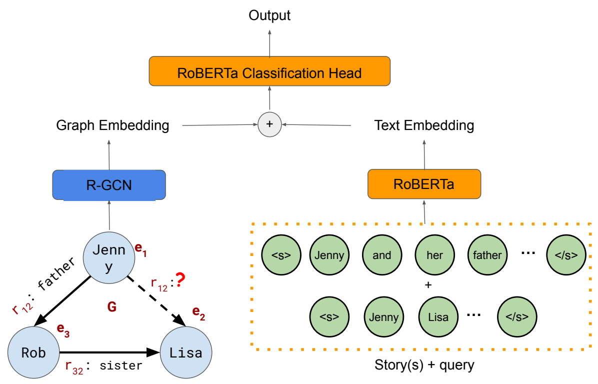


Figure 2: SSD model illustration: first obtain the graph embedding and text embedding separately using R-GCN and RoBERTa respectively, then adding the embeddings together and feeding through a classification layer to get the final output.

2 Problem Setup

Each example in CLUTTR (Sinha et al., 2019) is a tuple of the form (S, G, e_1, e_2), where S represents the story/passage describing the entities (fictional characters) and relations between them, G represents the kinship graph, e_1 and e_2 represent the pair of query entities (whose relationship is being queried). To aid clarity on these notations, we have illustrated the values of (S, G, e_1, e_2) corresponding to our running example in Figure 1. Further, each kinship graph can be considered to be a collection of entity nodes (E) and relation edges (R) (as illustrated in Figure 1), where $E = (e_1, e_2, e_3)$ and $R = (r_{12}, r_{13}, r_{32})$. Note that the kinship graph mentions only the relationships clearly stated in the story. For example, in Figure 1, the entity pairs (e_1, e_3) and (e_3, e_2) are explicitly mentioned in story S. The learning task is to predict the relationship between the two query entities. This is framed as a classification task over 20 possible relationship types in the dataset. The number of composition operations/steps required to infer the relationship between the query entities is denoted by k. For example, in figure 1, $k = 2$ for inferring the relationship between e_1 and e_2 as there are 2 composition operations needed to get the final result.

In this work, we empirically evaluate the utility of data augmentation and intermediate structured representations towards compositional generalization for the task of kinship prediction from a story. Next, we formally describe our model, SSD, where SSD stands for Systematic Compositional Generalization with Symbolic Representation and Data Augmentation for Kinship Prediction.

3 Method

We first describe our base model followed by a description of two strategies explored in this work - (1) data augmentation and (2) predicting and using intermediate kinship graphs.

Our base model, SSD (base) is adapted from the RoBERTa-based (Liu et al., 2019) baseline presented in Sinha et al. (2019). However, different from Sinha et al. (2019) we allow finetuning of the RoBERTa transformer layers. Grounding in the running example, given S, e_1, and e_2, SSD (base) predicts the relation r_{12} between e_1 and e_2 using the following three steps:

1. Obtaining story representation: This is the `[CLS]` representation by doing a forward pass of RoBERTa on the story, S.
2. Obtaining entity representations: During training, each entity (such as e_1, e_2, etc.) is replaced by a unique number in the story (following Sinha et al. (2019)). We obtain the representation for each entity by averaging the tokens from the last transformer layer of RoBERTa corresponding to the positions where the entity appeared in the story.
3. Classifier for predicting relation: This is a multi-class classification task (with total number of classes as the number of relationships possible in the dataset) using a linear classifier that takes as input the concatenation of representations of

the story and two query entities.

3.1 Data Augmentation

For each example in our training set, we augment the training set further by considering the pairs of entities for which the relation is explicitly mentioned in the story thus requiring no composition operations. We illustrate this data augmentation procedure using our running example in figure 1. We add the query entity pairs (e_1, e_3) and (e_2, e_3) in the training set as the relationships for these pair of entities are explicitly mentioned in the story. To predict the relation between the pair of entities mentioned in a query, the model has to operate in two stages, (1) extracting the relations mentioned explicitly in the story and (2) performing compositional reasoning over the extracted relations. This data augmentation procedure helps to ensure that the model becomes better at extracting the relations that are mentioned explicitly in the story, thus not propagating any error from the relation extraction stage to the compositional reasoning stage for predicting the target relation between the queried pair of entities. This model is denoted as `SSD` (data aug) henceforth. For inference using `SSD` (data aug) one needs to provide all the pairs of query entities whose relations can be extracted directly from the text of the story in addition to the actual pair of query entities.

3.2 Intermediate Kinship Graphs

Prior work has found models using structured representation of stories in form of kinship graphs perform better than transformer models trained only on stories for this task. However, it is unreasonable to assume that we will always have access to gold kinship graphs for the task of kinship prediction from narratives or stories during inference. Hence, we empirically evaluate the utility of predicting an intermediate kinship graph and then jointly reasoning over the predicted graph and the input story to predict the relation between the queried pair of entities. We illustrate our strategy using the running example in figure 1. We form the intermediate kinship graph, G' by predicting the relations between the entities whose relations are explicitly mentioned in the story. We predict the relations to form this intermediate graph by using a linear layer over representations of the story and the pair of query entities obtained using a RoBERTa model.

Next, we obtain two representations of the target relation based on (1) text: using linear layer over representations of the story and the pair of query entities obtained using a RoBERTa model and (2) graph: using linear layer over representations of kinship graph and query entities obtained using R-GCN (Schlichtkrull et al., 2017) (see Appendix for details). We concatenate these two target relation representations and use another linear layer to predict the target relation. This model is denoted as `SSD` (graph) henceforth.

Similar to `SSD` (data aug), for inference using `SSD` (graph) one needs to provide all the pairs of query entities whose relations can be extracted directly from the text of the story in addition to the actual pair of query entities.

4 Experiments and Results

All models are trained using cross-entropy loss. Every model is trained with 40 epochs and a learning rate of 5e-6.

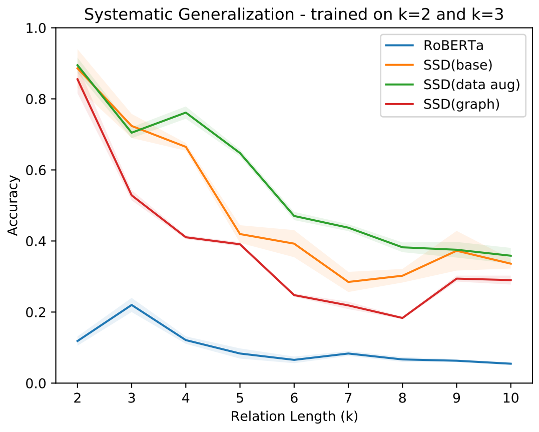
4.1 Baseline and Evaluation Metrics

We consider the RoBERTa-based model in Sinha et al. (2019) as our baseline. Note that in the baseline the transformer layers of RoBERTa are not finetuned. For all our experiments we report the accuracy of predicting the relation between the queried pair of entities. Further, following Sinha et al. (2019), we report the accuracy over multiple test sets, where each test set is characterized by k, the number of composition operations/steps required to find the relation between the queried pair of entities. For example, in figure 1, the number of composition steps (k) is 2. In test sets of CLUTRR, k varies from 2 to 10.

4.2 Evaluating compositional generalization

Figure 3 shows the accuracy of different variants of `SSD` on the test sets of CLUTRR. We consider two settings, where `SSD` is trained on data with (1) $k = 2, 3$ and (2) $k = 2, 3, 4$ Irrespective of the training data complexity (in terms of k), we observe that `SSD` (data aug) outperforms baseline. Notably, we see improvements even when $k = 10$ during test showing the utility of data augmentation for improving the generalizability of the models.

While data augmentation shows promise, we do not see any improvements when predicting and reasoning jointly over the intermediate kinship graph. Rather, the performance of the models drop significantly when we predict the relation conditioned on the story and the intermediate kinship graph. This



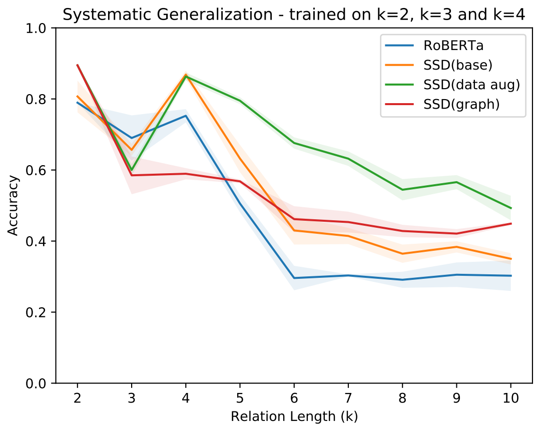


Figure 3: compositional generalization performance of different models when trained on $k = 2, 3$ and $k = 2, 3, 4$. Our presented strategies boost accuracy even when the number of composition steps (k) is 10.

is counter-intuitive as we hypothesized the intermediate kinship graph (which is structured) would aid the model further in making compositions. As one of the possible reasons for this, we hypothesize that our method of fusing representations from two modalities, story and graph, might be sub-optimal that results in the failure. Future work can explicitly look into devising better techniques for this fusion.

Generalization with noisy inputs: We also evaluate the models with noisy train and noisy test sets of CLUTRR following Sinha et al. (2019). We explore the following three noisy data settings shown in Figure 4:

- Supporting facts: There are two reasoning paths that can lead to the correct answer p_c and p_n. These two paths has the same beginning and ending nodes but p_c is shorter than p_n (smaller k).

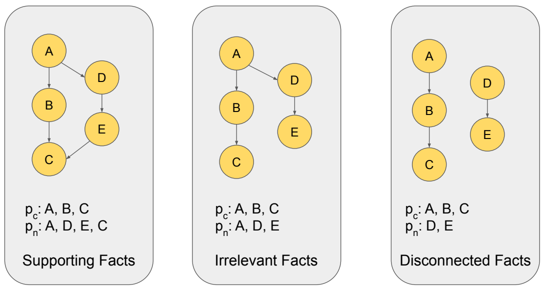


Figure 4: Categories of Noisy Inputs. The query is finding the relationship between entity A and entity C.

- Irrelevant facts: p_n, the path that contains the irrelevant facts, shares the same beginning node with p_c which leads to the correct answer. p_n can be seen as a branch of the graph that doesn't lead to the correct answer.
- Disconnected facts: p_n, which is the path that contains the disconnected facts, can be treated as another graph that is disconnected from the main story that contains the reasoning path p_c, which leads to the correct answer.

Table 1 shows the result of different SSD variants when evaluated on the noisy test sets. The model performance decreases as the number of deduction steps required (k) increases, which is consistent with other experiments' results. We can also notice that the models, SSD (base) and SSD (graph), tend to perform better with graphs that contain supporting facts, irrelevant facts, and disconnected facts compared to graphs that are free of noise but require the same number of composition operations (k) to predict the target relation. This shows that SSD is good at identifying useful and relevant information from the graph and extra information from the noisy inputs improves the models' performance.

4.3 Varying the amount of additional annotation

For data augmentation and also for predicting the intermediate kinship graphs we need additional annotation to identify entity pairs whose relationship is explicitly mentioned in the text. While there can be heuristic approaches to estimate such entity pairs (for example, set of all distinct entity pairs that appear in the same sentence), in this work we re-purpose the gold kinship graphs to get this annotation. Realistically, having gold kinship graphs

Model	Train Set	Test Set	Accuracy	Test Set	Accuracy	Test Set	Accuracy	Test Set	Accuracy	Test Set	Accuracy	Average Accuracy
SSD (base)	1.2,1.3	1.2	82.2%	1.3	61.0%	2.3	75.9%	3.3	76.5%	4.3	69.6%	73.0%
	2.2,2.3	2.2	93.4%	2.3	84.1%	1.3	61.3%	3.3	75.9%	4.3	72.4%	77.4%
	3.2,3.3	3.2	97.4%	3.3	72.4%	1.3	56.5%	2.3	79.9%	4.3	69.6%	75.2%
	4.2,4.3	4.2	62.5%	4.3	68.9%	1.3	55.3%	2.3	77.1%	3.3	72.9%	67.3%
SSD (graph)	1.2,1.3	1.2	88.8%	1.3	43.3%	2.3	69.6%	3.3	69.5%	4.3	58.3%	65.9%
	2.2,2.3	2.2	92.1%	2.3	75.7%	1.3	42.7%	3.3	70.8%	4.3	54.1%	67.1%
	3.2,3.3	3.2	97.4%	3.3	64.3%	1.3	36.8%	2.3	71.9%	4.3	50.0%	64.1%
	4.2,4.3	4.2	68.4%	4.3	58.9%	1.3	43.3%	2.3	72.4%	3.3	69.3%	62.5%

Table 1: Testing SSD (base) and SSD (graph) performance when training on story graphs with or without noisy inputs. The integer after symbol . represents the number of steps required to infer the relationship between the query entities, which is k as mentioned section 2.1, and the integer before the symbol . has the following meaning provided by the original CLUTRR paper (Sinha et al., 2019): 1=free of noise; 2=with supporting facts; 3 = with irrelevant facts; 4 = with disconnected facts.

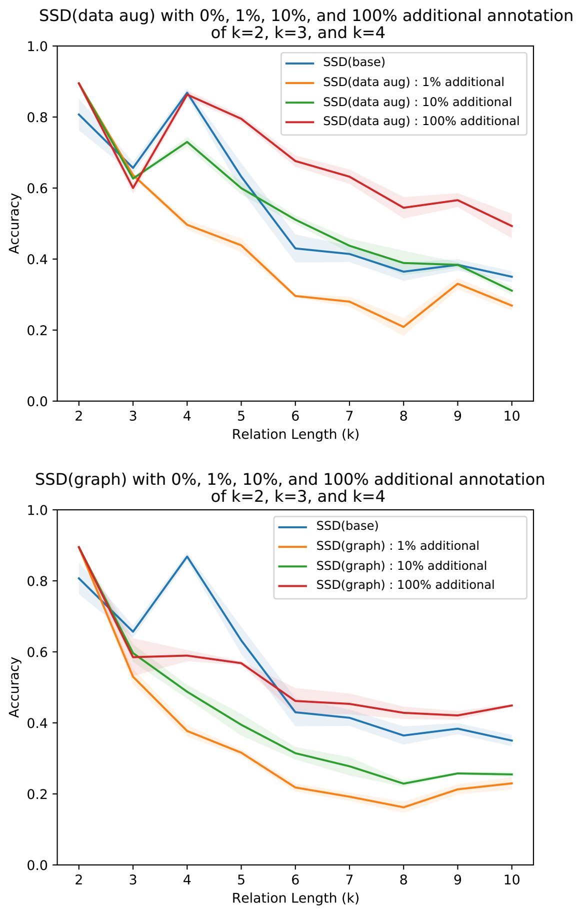


Figure 5: Comparison of model performance when additional supervision (through data augmentation and intermediate kinship graphs) is only available for 1% and 10% of the data and the rest is trained without additional supervision.

for all the training data might not be feasible. In this section we empirically explore how much performance improvement we would achieve if we had access to only 1% (and 10%) of gold kinship graphs to obtain the additional annotation of entity pairs for data augmentation.

Figure 5 our assumption is reasonable as the performance of only allowing additional supervision for 10% of the training data achieves decent accuracy.

4.4 Low data regime

Next, we study the effect of reducing the size of training dataset and evaluate the effectiveness of our strategies under this setting. We reduce the training data size gradually by an order of 10 and form two smaller training splits with sizes around 1000 and 100 samples. Figure 6 and Figure 7 shows the results of our proposed model on the standard (no-noise) CLUTRR test datasets as we reduce the overall size of our training datasets with k=2,3 and k=2,3,4 respectively. We find that even in low data regime data augmentation leads to improvements.

5 Conclusion

In this paper, we present SSD to empirically evaluate the utility of two strategies (1) data augmentation and (2) predicting and using intermediate kinship graphs, towards compositional generalization of transformer-based models for the task of kinship prediction from a story. While data augmentation boosts the performance of our model, using intermediate kinship graphs leads to a downfall in the overall performance. Data augmentation is fruitful even when additional supervision in form of ground-truth kinship graphs is present for a limited set of examples. Future work can explore better methods to fuse the information from the intermediate kinship graph and the story instead of simple concatenation as done in this work.

References

Dzmitry Bahdanau, Shikhar Murty, Michael Noukhovitch, Thien Huu Nguyen, Harm de Vries, and Aaron Courville. 2018. Systematic generaliza-

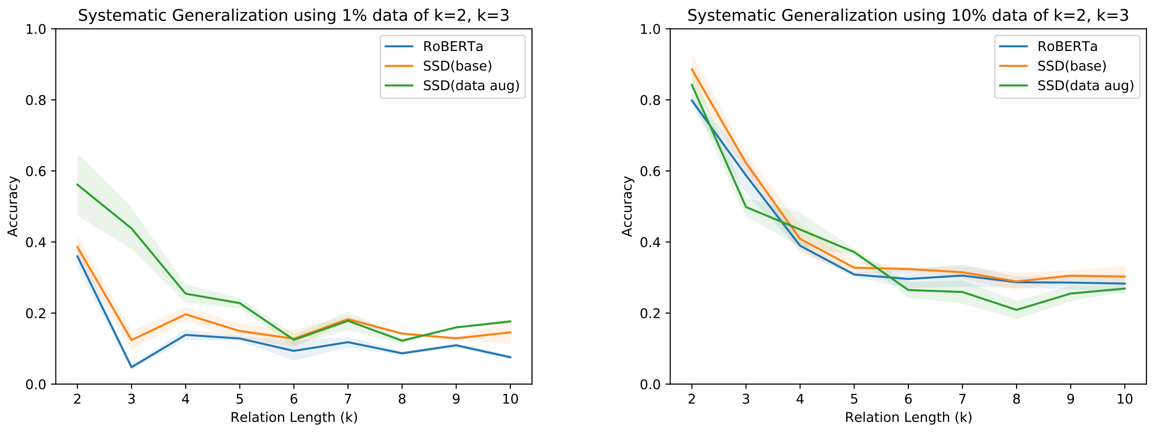


Figure 6: Low data regime performance of settings for RoBERTa when trained on k=2,3. Use of augmented data from the ground-truth kinship graph boosts accuracy even when the overall size of the training data is reduced.

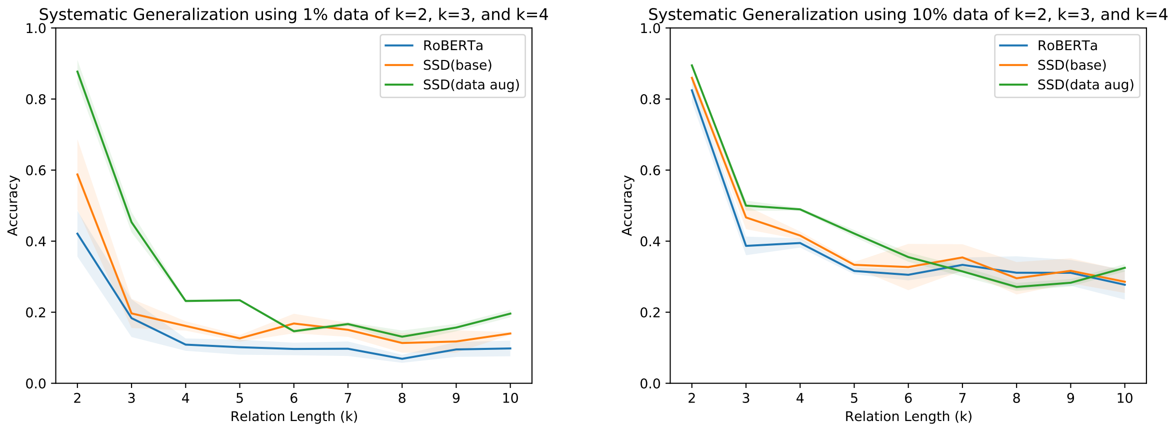


Figure 7: Low data regime performance of settings for RoBERTa when trained on k=2,3,4. Use of augmented data from the ground-truth kinship graph boosts accuracy even when the overall size of the training data is reduced.

tion: what is required and can it be learned? *arXiv preprint arXiv:1811.12889*.

Aloka Fernando and Surangika Ranathunga. 2022. Data augmentation to address out-of-vocabulary problem in low-resource sinhala-english neural machine translation.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R Bowman, and Noah A Smith. 2018. Annotation artifacts in natural language inference data. *arXiv preprint arXiv:1803.02324*.

Divyansh Kaushik and Zachary C Lipton. 2018. How much reading does reading comprehension require? a critical investigation of popular benchmarks. *arXiv preprint arXiv:1808.04926*.

Brenden Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International conference on machine learning*, pages 2873–2882. PMLR.

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.

Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2017. Modeling relational data with graph convolutional networks.

Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L. Hamilton. 2019. CLUTRR: A diagnostic benchmark for inductive reasoning from text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4506–4515, Hong Kong, China. Association for Computational Linguistics.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Highway networks. *CoRR*, abs/1505.00387.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

David Wan and Mohit Bansal. 2022. Factpegasus: Factuality-aware pre-training and fine-tuning for abstractive summarization.

Kevin Yang, Olivia Deng, Charles Chen, Richard Shin, Subhro Roy, and Benjamin Van Durme. 2022. Addressing resource and privacy constraints in semantic parsing through data augmentation.

Appendix

A Description of models used to encode and reason over the intermediate kinship graph

A.1 R-GCN

The formula for Relational-Graph Conventional Networks we used is:

$$h_i^{l+1} = \sigma(W_0^l h_i^l + \sum_{r \in R} \sum_{j \in N_i^r} \frac{1}{C_i^r} W_r^l h_j^l) \tag{1}$$

where $W_0^l h_i^l$ gives special treatment to self connection, r represents the relation type, j represents the neighbor nodes of node i with relation r, and W_r^l is the projection matrix for each relation type. In our setting, we have three R-GCN (Schlichtkrull et al., 2017) layers. h is the hidden representation of an entity in the graph and r is a kinship-relation type that belongs to set R, which contains all possible relations.

A.2 Highway Connection

We utilize highway connections (Srivastava et al., 2015) between R-GCN (Schlichtkrull et al., 2017)layers:

$$g = Sigmoid(W_{hw}(\hat{h_i})) \tag{2a}$$

$$h_i^{l+1} = g \odot \hat{h_i} + (1-g) \odot h_i^l \tag{2b}$$

where W_{hw} is a linear layer, and $\odot$ denotes element-wise multiplication. h_i^l is the entity representation of the nodes in the graph from the previous layer, and $\hat{h_i}$ is the entity representation of the node in the graph acquire by passing h_i^l to a R-GCN (Schlichtkrull et al., 2017) layer.