Understanding LLM's Symbolic Reasoning Through Comparative Analysis of DSR Methodology

Abhinav Gupta (agupta67@usc.edu)
Akshita Kapur (kapuraks@usc.edu)
Dhruvam Zaveri (dzaveri@usc.edu)
Hrishikesh Thakur (hthakur@usc.edu)
Shreyas Shrawage (shrawage@usc.edu)

Abstract

In this project, we compared the performances of four different models - RoBERTa, XLNET, ALBERT, and ELECTRA - on the logical reasoning task of kinship classification. Using the Differential Symbolic Reasoning (DSR) Framework, trained using CLUTRR dataset. Through four key stages including dataset preprocessing, relationship extraction, logical rule application, and final output prediction, we implemented the DSR-LM architecture with adaptable modifications to suit various classifiers. The results indicate a significant improvement in performance with the integration of the DSR module, showing an increase of 75% on average in testing accuracy across all four classifiers, highlighting the efficacy of the DSR-LM approach in enhancing LLM logical reasoning.

1 Introduction

Large Language Models (LLMs) have revolutionized natural language processing, enabling dynamic problem-solving capabilities in real-time. Models like ChatGPT leverage vast amounts of training data to generate coherent responses to user queries. However, tackling niche domains and logical questions still presents a complex challenge. Differential Symbolic Reasoning (DSR-LM) emerges as a solution, enhancing logical reasoning by integrating pre-trained models with a symbolic reasoning module.

1.1 DSR-LM Framework

The framework extracts textual relationships, applies logical rules, autonomously learns, and refine these rules. By incorporating semantic loss and integrity constraints, DSR-LM ensures logical consistency while balancing deduction and semantic losses through a reasoning engine and a two-part loss function. Leveraging Scallop, a neurosymbolic platform blending deep learning and logical

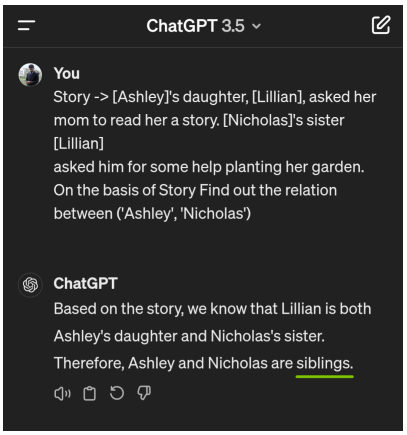

Figure 1: ChatGPT Example

reasoning, DSR-LM enhances accuracy, efficiency, interpretability, and generalizability over conventional methods.

1.2 Project Overview

This project aims to extend the DSR-LM framework beyond its original implementation with RoBERTa. Building upon this foundation, we integrated the DSR module with XLNET, ALBERT, and ELECTRA. The models are trained to perform the task of kinship classification on the CLUTRR dataset that consists of semi-synthetic stories with complex family relationships. The dataset is also augmented by 92 manually crafted kinship composition rules. The project implements the DSR-LM architecture across the chosen LLMs, through four stages: dataset pre-processing, relationship extraction, logical rule application (fact extraction), and output prediction based on deduced facts.

2 Related Work

DSR provides a robust framework for learning interpretable logic rules by generating and refining these rules through gradient-based optimization, significantly enhancing the learning process (Zhang, 2023). Scallop, utilized for differentiable

symbolic inference, processes relation types and deduces kinship via deductive rules while managing integrity constraints, which improves the model's ability to handle complex logical reasoning (Zhang, 2023).

For kinship identification, we used models like RoBERTa, ALBERT, XLNet, and ELECTRA to take advantage of the advanced language understanding capabilities they possess. RoBERTa's deep contextual training is crucial for complex kinship analysis (Liu Y., 2019). ALBERT offers robust performance with fewer parameters, useful in diverse scenarios (Lan Z., 2019). XLNet's bidirectional context capture aids in nuanced relation parsing (Yang Z., 2019). ELECTRA's discriminative training effectively handles subtleties in kinship terms, enhancing network mapping accuracy (Clark K., 2020).

3 Problem Description

Commercial LLMs like ChatGPT and Gemini are very good at solving dynamic problems in real-time. However, they still struggle at solving logical reasoning tasks. For example, refer figure 1. We asked ChatGPT to derive the relationship between 2 people given a short description of relationships between 3 people. Relationship defined as A is related to B as X, and B is related to C as Y. Then what is the relation between A and C? ChatGPT3.5 clearly fails to answer it correctly, and predicts a mother-son relationship as siblings. This shows the vast scope of improvement even in advanced LLMs.

One reason for this limitation is that LLMs are heavily trained using multiple supervised and unsupervised learning techniques. For some tasks, they are given a desired output format, whereas for others, the model only gets dumps of data and is expected to learn patterns from it. Therefore, most of the learning is example based. The model is only able to predict one-word at a time based on the highest probability and similarity index. There is no real-reasoning to it. Therefore, in this project, we attempted to add a reasoning module with the aim of one task - kinship classification.

4 Methods

4.1 Libraries

For our project, we are using Python and several specialized libraries to enhance our capabilities in data handling and Natural Language Processing:

- **Scallopy:** We employ Scallop through Scallopy, a python library, for logical reasoning. Scallop dynamically infers and validates relationships using differentiable logic.

- **Transformers Library:** Provided by Hugging Face, it grants access to a variety of pre-trained models, including RoBERTa, XLNET, ELECTRA, and ALBERT, which we fine-tune for text classification. The library also includes vital tokenizers that prepare text data for processing by their respective models.

- **PyTorch:** PyTorch integrates seamlessly with Scallop for symbolic inference. Its flexibility allows us to build neural network models (MLP) used for relationship classification.

4.2 Dataset

For our project, we utilize the CLUTRR dataset, specifically focusing on certain key columns and employing distinct preprocessing steps tailored to our baseline and DSR-LM models. Considering the computational limitation of our system, our final train-test dataset sizes are 2000 and 500 respectively, maintaining the 80-20 ratio. Refer to table 1, for a more detailed breakdown about each component of the CLUTTR dataset used in the models.

Column	Description
story	Semi-synthetic story with hypothetical families
query	Target query with two names, goal is to classify kinship between the two entities
target	Indicator of correct relation in query
target text	Text for the correct relation for the query. The indicator follows the rule as follows: "aunt": 0, "son-in-law": 1, ... "sister-in-law": 20
genders	Genders of names in the story
task split	Train, Test

Table 1: Dataset Component Breakdown

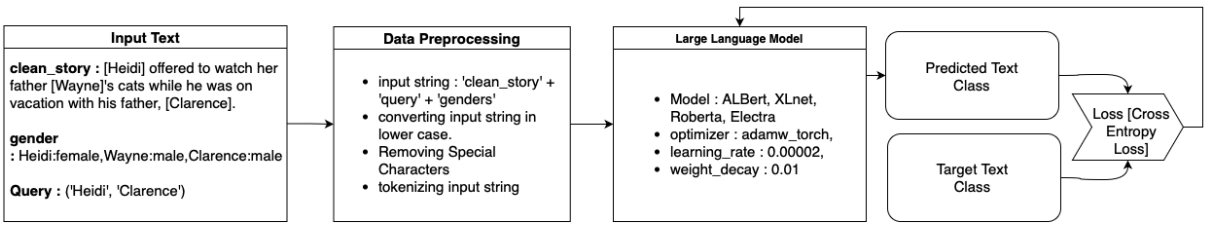

Figure 2: Baseline Model Architecture and Text Data Flow through the Model Components

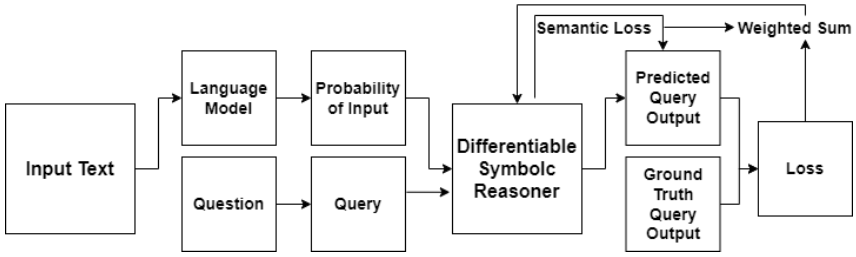

Figure 3: DSR-LM Model Architecture Diagram with Flow of Components

Model	w/o DSR	w/ DSR
ALBERT	19.125%	37.0%
RoBERTa	23.6%	35.6%
XLNet	13.75%	21.5%
ELECTRA	18.75%	37.5%

Table 2: Final Accuracies

4.3 Model Architecture

4.3.1 Baseline Model:

Preprocessing Steps: Concatenation of Inputs: The columns 'clean_story', 'query', and 'genders' are concatenated to form a single input string, providing all necessary context for the model to predict the relationship.

Text Cleaning: Convert all text to lowercase to ensure consistency and to prevent case-sensitive discrepancies, and use regular expressions to remove special characters such as square brackets, parentheses and single quotes to standardize the text, prevent parsing errors.

Referring to figure 2, our model architecture shows that this processed data is fed as tokenized tensors to pre-trained hugging face text classifiers selected in our project , to get the output relationship.

4.3.2 DSR-LM Model:

Parsing and Tokenization: Stories are split into individual sentences, each treated as a separate context unit to facilitate detailed analysis. Queries are used to extract subject and object entities for relationship classification.

Context Preparation: Extracted sentences are stored and handled as distinct context elements, each providing specific information about the relationships between characters. Contexts are preprocessed to clean and organize text efficiently, with sentences possibly concatenated based on relevance and the presence of named entities for comprehensive relation extraction.

Extract Relations: Pre-trained LLM is used to extract embeddings from the pre-processed tokenized text and MLP is used to extract relations using these embeddings.

Extract Facts: Relationship labels (answers) are converted into numeric IDs based on a predefined map, facilitating model processing and output interpretation.

Referring to figure 3, our model architecture shows that LLM assigns probabilities to map pairs of individuals to all the relationships and DSR updates the probabilities and outputs the relationship that has the highest probability.

4.4 Text-Classifiers Selected:

We chose RoBERTa, XLNet, ALBERT, and ELECTRA for the following reasons:

- **RoBERTa** to draw a comparison with the reference paper.
- **ALBERT** provides high efficiency with fewer parameters, making it suitable for tasks requiring complex reasoning without extensive computational resources.

- **XLNet** is adept at processing long-range dependencies and contextual nuances, essential for logical reasoning.
- **ELECTRA** excels at distinguishing subtle differences in text, a key ability for precise logical analysis.

5 Experimental Results

In our project, we selected RoBERTa as one of the text classifiers specifically to include it in our study, as we aimed to conduct a comparative analysis against the reference paper that guided our project (Zhang, 2023). This paper implemented a DSR layer on RoBERTa (DSR-Roberta) and demonstrated a test accuracy of 60.98% when trained on the entire CLUTTER dataset of 12,000 data points. Moreover, the baseline model of RoBERTa has a test accuracy of 34.8%.

From table 2, we can note that the test accuracy recorded for the DSR-RoBERTa model developed by us is 35.6% and the baseline model's test accuracy is 23.6%. Important to note here that the difference in results is largely due to the scale of the training data; our model has seen merely 2,000 stories (due to computational constraints) during training, whereas the models from the reference paper were trained on 12,000 rows.

This shows that using just one-sixth of the training data, we achieved relatively high accuracy, mirroring the performance trend of the reference paper. Our results improved with larger datasets, confirming that our model aligns well with established benchmarks.

With a solid foundation established by our DSR-RoBERTa model, we can now effectively compare the performance of other DSR-LM models. For a fair and accurate assessment, each DSR-LM should be compared only to its respective baseline counterpart.

Referring to table 2 and figure 4, models enhanced with a DSR layer showed a 75% average increase in accuracy over baseline models. Both ALBERT and ELECTRA performed comparably to RoBERTa, highlighting their effectiveness for tasks involving local context and entity relationships. XLNET under-performed due to its design for longer texts, which did not suit our short paragraph inputs, leading to lower results. However, integrating the DSR framework significantly improved XLNET's logical reasoning, indicating its potential for more suitable tasks.

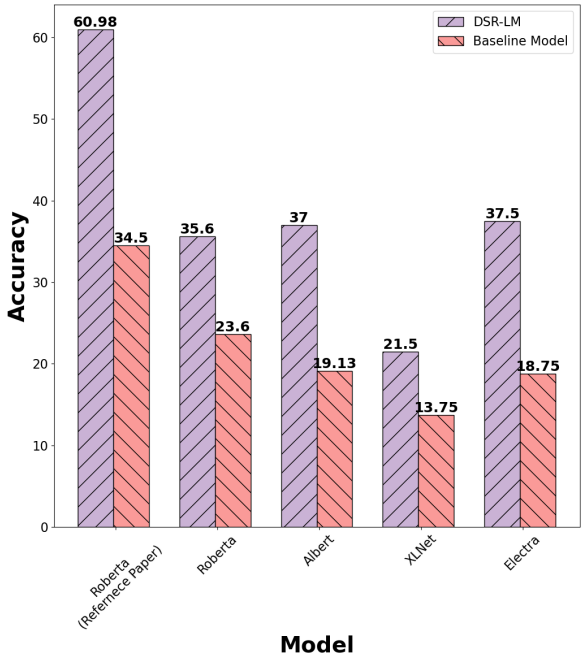

Figure 4: Graph Comparing Performance of Self-Developed DSR-LMs vs Reference Paper

6 Conclusions and Future Work

To conclude, it is evident from the research conducted that the DSR-framework is a powerful tool that can significantly enhance the logical-reasoning capabilities of language models for text classification. Advancing the existing research beyond the use of a single-focus text classifier like RoBERTa, through this project, we were able to demonstrate that there are alternative text classification LMs that can be just as effective. Based on our findings, we can positively report that these models have the potential to outperform or at least match the performance of current bench-mark set by the DSR-RoBERTa model.

Moving forward, expanding the framework to encompass tasks beyond kinship-classification will diversify the logical reasoning capabilities of LMs. Given the optimization of LMs for specific applications, employing different LMs for varied tasks, as demonstrated in this project, is crucial. Increased adaptation will yield more robust and versatile models, enhancing their capacity to tackle a wider array of logical reasoning challenges. These advancements hold significant promise for Natural Language Processing, amplifying the practical applications and effectiveness of language models in the modern world.

7 Division of Labor

- **Abhinav Gupta**: project outline, framework design, and modeling, literature review, baseline model development, fine-tuning of DSR-LM models, report work
- **Akshita Kapur**: framework design and modeling, literature review, dataset preparation and pre-processing, DSR-LM model development, report work
- **Dhruvam Zaveri**: literature review, dataset preparation, and pre-processing, baseline model development, deriving results and conclusion, report work
- **Hrishikesh Thakur**: framework design and modeling, DSR-LM architecture and model development, fine-tuning of DSR-LM models, final result compilation and review, report work
- **Shreyas Shrawage**: project outline, literature review, baseline model development, DSR-LM model development, deriving results and conclusion, report work

References

Le Q. V. Manning C. D. Clark K., Luong M. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *ArXiv. /abs/2003.10555*.

Goodman S. Gimpel K. Sharma P. Soricut R. Lan Z., Chen M. 2019. Albert: A lite bert for self-supervised learning of language representations. *ArXiv. /abs/1909.11942*.

Goyal N. Du J. Joshi M. Chen D. Levy O. Lewis M. Zettlemoyer L. Stoyanov V. Liu Y., Ott M. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv. /abs/1907.11692*.

Yang Y. Carbonell J. Salakhutdinov R. Le Q. V. Yang Z., Dai Z. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *ArXiv. /abs/1906.08237*.

Huang J. Li Z. Naik M. Xing E Zhang, H. 2023. Improved logical reasoning of language models via differentiable symbolic programming.