Neuro-Symbolic Methods For Natural Language Inference and Question Answering

by

Yufei Feng

A thesis submitted to the
Department of Electrical and Computer Engineering
in conformity with the requirements for
the degree of Doctor of Philosophy

Queen's University
Kingston, Ontario, Canada
October 2022

Abstract

One of the fundamental problems in deep learning research is how to design neural network models to incorporate logic and symbolic operations. Although deep neural network models have achieved state-of-the-art performance on multiple natural language processing benchmarks, those black-box models can hardly provide explanations for their inner mechanisms. They still lack the ability to perform systematic reasoning like human beings and generalize poorly to out-of-distribution samples.

In this thesis, we attempt to overcome these limitations by designing neuro-symbolic models that combine neural network models with natural logic theory. At the lower level, we use neural networks to model the text representation and produce intermediate predictions, while at the higher level, we leverage symbolic operations to perform reasoning, which leads to the final prediction. We apply our neuro-symbolic models to solve the natural language inference (NLI) task, and we also explore ways to extend our method to question answering (QA).

This thesis offers a set of contributions that address the problem of effectively combining deep neural networks with symbolic methods. The first contribution is a novel end-to-end differentiable natural logic model for NLI. Our proposed model achieves empirically competitive results on the Stanford NLI benchmark and multiple stress-test datasets. Our model also provides faithful explanations for its decisions based on

natural logic. The second contribution is a novel neuro-symbolic NLI model, which overcomes the limitations of its predecessor by leveraging a well-designed natural logic program and reinforcement learning. We also propose an introspective revision algorithm that incorporates commonsense knowledge bases to alleviate the spurious reasoning problem and improve training efficiency. The third contribution is an extension of the neuro-symbolic method to multi-hop QA applications. We propose a model that accurately locates chains of useful evidence, which can be trained without direct supervision, and a neuro-symbolic QA model that performs natural-logic style reasoning on the chains of evidence.

Acknowledgments

I want to express sincere thanks to Xiaodan Zhu, Michael Greenspan, Ali Etemad, Yanfei Liu, Steven Ding, Leila Kosseim, Cao Thang Dinh, Xiaoyu Yang, Mo Yu, Xiaoxiao Guo, Zhan Shi, Yuping Ruan, Hui Liu, Jiacheng Gu, Feng Nie, Zi'ou Zheng, Tianda Li, Quan Liu, Rohan Bhambhoria, Stephen Obadinma, Debra Fraser, Cheryl Wright, Rose M. Silva and all my family members in China for their invaluable help to my Ph.D. study at Queen's University.

Wish all the good people safe and happy.

Statement of Originality

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Deep neural networks currently show great strength in modeling and recognizing patterns within natural language and images. Facilitated by the development of large annotated datasets and well-designed neural network architectures, deep learning models have achieved impressive performance in applications such as text classification (Socher et al., 2013; Bowman et al., 2015; Wang et al., 2018a), reading comprehension (Rajpurkar et al., 2016; Welbl et al., 2018; Yang et al., 2018), summarization (Nallapati et al., 2016), and information extraction (Angeli et al., 2015; Stanovsky et al., 2018; Kolluru et al., 2020), etc.

Typically supervised neural network models learn to perform specific tasks by fitting large labeled datasets in an end-to-end manner. The network parameters are updated by performing gradient descent, which maximizes the designed objective function or the data likelihood. Compared to previous statistical learning models (Hastie et al., 2009), such a learning paradigm has many advantages when dealing with complex data such as natural language sentences or images. Among the most

prominent features, a neural network can contain billions of parameters, which enables the model to approximate highly complex functions. Instead of handcrafting input features, the end-to-end training method can discover useful and task-specific features automatically through back-propagation, and most importantly, deep neural networks have the benefit of a distributed representation (Hinton, 1984), which greatly advances the research of artificial intelligence.

Challenges of End-to-End Deep Neural Networks. Although deep learning models achieve state-of-the-art performance on multiple task benchmarks (Wang et al., 2018a; Devlin et al., 2019a; Liu et al., 2019), several fundamental problems remain:

- Deep neural networks are data-hungry, and good performance is often achieved by fitting large labeled datasets. The high-capacity models over-fitted to relatively limited training resources may generalize poorly to out-of-distribution data.

- Deep neural networks are black-box models that provide little explanation for their inner mechanism. With distributed representation, the decision boundary lies in the space of hundreds of dimensions. The bases (axes) of the space are latent variables, which cannot be easily interpreted.

- In many circumstances, the learning process can hardly be controlled. It is particularly hard to make neural networks learn according to existing human logic, such as first-order logic and natural logic (MacCartney, 2009). Meanwhile, even preventing the model from fitting to a known dataset bias may require extensive effort.

Neuro-Symbolic Method. Part of the weakness of neural networks, however, is the strength of the symbolic systems. Based on logic and linguistic theories, symbolic systems often endow built-in explainability and are less data-hungry, though they can be vulnerable to noise and variations in the data. To combine the strength of the neural networks and the symbolic systems, neuro-symbolic models (Garcez et al., 2002, 2008, 2015, 2019; Mao et al., 2019; De Raedt et al., 2019) have been designed to deal with the aforementioned challenges. While using a neural network to extract features from the samples and model lower-level perceptions such as object recognition and token representation, at the high level, a typical model uses symbolic operators to perform required reasoning. For example, the neural symbolic concept learner model (Mao et al., 2019) uses a neural network to recognize shapes and colors (Johnson et al., 2017), while at the higher level it uses executable neural programs to perform tasks such as counting and comparison. The high-level symbolic programs, being different depending on the specific tasks, serve as the reasoning evidence, which *faithfully* reveals how the model derives the final prediction.

Given the variety of the neuro-symbolic methods, in this thesis, we focus on neuro-symbolic models that are designed for natural language understanding, and during training, the model only consumes input data and the final answer, without having access to the ground-truth logic programs and their intermediate outputs. The neuro-symbolic model is trained by back-propagation and the gradient descent method. Building such a neuro-symbolic system faces several challenges:

- How to design the neural symbolic framework to work efficiently in an end-to-end manner, even if the logic operations may not be differentiable? A neuro-symbolic model needs to learn how to discover useful features and how to perform reasoning through back-propagation.

- The symbolic reasoning process varies depending on the tasks and applications. How to design domain-specific logic programs that scale up to the available data becomes a central problem.

- Owing to the annotation expenses, we often cannot expect detailed human annotations for the ground-truth symbolic programs or their intermediate outputs. How can we prevent spurious logic programs that may execute to the correct final answer in an incorrect or degenerate way?

Our Methods and Applications. Motivated by the strengths and challenges of neuro-symbolic models (Rocktäschel and Riedel, 2017; Mao et al., 2019; Khot et al., 2020), in this thesis we develop neuro-symbolic models for natural language inference (NLI) (Cooper et al., 1996; Bentivogli et al., 2009; Bowman et al., 2015; Williams, 1992; Marelli et al., 2014) and multi-hop question answering (QA) (Mishra et al., 2021; Yang et al., 2018; Welbl et al., 2018; Thorne et al., 2018; Chen et al., 2019a; Fang et al., 2020). NLI is a fundamental text reasoning task in natural language processing, which requires algorithms to determine the inference relation between two short, ordered texts (MacCartney and Manning, 2008), and multi-hop QA requires models to gather information from different parts of a text to answer a question (Chen et al., 2019a). We show in Figure 1.1 an example of an NLI task and an example of the multi-hop QA, and the detailed task definition can be found in Section 2. To

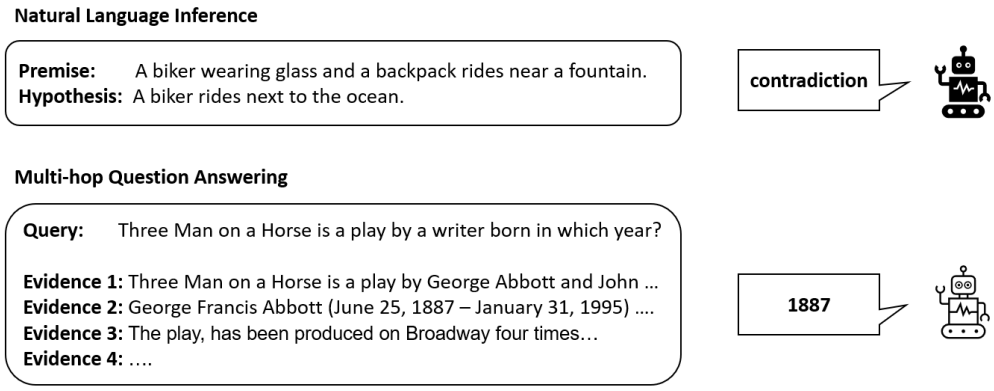

Figure 1.1: Examples for natural language inference and multi-hop question answering

solve the NLI problem, natural logic (Lakoff, 1970; van Benthem, 1988; Valencia, 1991; Van Benthem, 1995; Nairn et al., 2006; MacCartney, 2009; MacCartney and Manning, 2009; Icard, 2012; Angeli and Manning, 2014), which will be detailed in Section 2.4, is proposed as a self-contained proof system that justifies the inference relations. Unlike neural network models, which learn to reason by training on large annotated datasets, natural logic is a rule-based system that functions by querying knowledge bases or expert systems. As a result, natural logic can be brittle to noise and linguistic variations. In this thesis, we propose to combine natural logic theory and the end-to-end differentiable neural network: the model trains and predicts like a neural network, but rather than being a black box, the model reasons according to natural logic. We further explore how key features of natural logic, which include stepwise reasoning and surface-form reasoning (detailed in Section 2.4), can be applied to the multi-hop QA task.

1.2 Thesis Statement

In this thesis, we show that an effective neuro-symbolic model can be developed to combine neural networks with natural logic. The resulting neuro-symbolic natural language inference (NLI) system proves to have both competitive empirical performance and interpretability. The neuro-symbolic NLI model can be enhanced by the refined natural logic symbolic program and reinforcement learning, while external commonsense knowledge bases can further improve the symbolic reasoning process. We also show that the step-wise reasoning of natural logic can be extended to the multi-hop question answering (QA) task, and with properly designed evidence retrieval and text reasoning methods, the resulting model achieves state-of-the-art performance.

1.3 Contributions

Combine natural logic and neural network and design neuro-symbolic models for natural language inference. Instead of directly applying black-box neural network models to large annotated datasets, we explore ways to build neural network models that follow a well-formulated natural logic theory (MacCartney, 2009). To achieve this goal, we reformulate natural logic as a state transition process and specify the probability model behind it. We then design and implement the first neuro-symbolic model for NLI (Section 3). We designed benchmarks by collecting multiple recently proposed stress-test NLI datasets and by building an extra dataset to evaluate the natural-logic-based interpretability. We conduct experiments to show that our method simultaneously achieves competitive performance on both standard benchmarks, that is, Bowman et al. (2015) and various linguistic probes (Yanaka et al., 2019a,b, 2020). Meanwhile, our method provides faithful explanations for its

inner-working mechanism. [1]

Improving neuro-symbolic natural language inference (NLI) with natural logic programs, reinforcement learning, and external commonsense knowledge base. Based on the analysis of the neuro-symbolic model proposed in Section 3, we advance by introducing a natural logic program, which takes the intermediate results generated by the neural network as input and executes to obtain the final prediction. We show that we can train a network to support the natural logic program reasoning by reinforcement learning. We also show that the advanced model provides an interface for leveraging commonsense knowledge bases, which improves the training efficiency and alleviates spurious reasoning. The advanced model empirically exceeds the performance of its predecessor by a large margin on both NLI performance and interpretability. [2]

[1] The contribution of this part of research corresponds to our publication Feng et al. (2020). The original idea for the proposed model and its key components were formed by the first, second, and the last two authors (the co-supervisors). The last author, Xiaodan Zhu, is the principal supervisor of this research. I proposed and implemented the following key components of the proposed framework: memory-enhanced modular network, the marginal likelihood framework and hard EM component and the integration of them with the loss function, implementation of the tree-based aggregation model, and the last two lexical constraints. I ran the experiments, tuned the models, and obtained all the results. I proposed and generated the 2-hop evaluation dataset. I designed the evaluation for aggregation steps and performed quantitative evaluation for interpretability. I implemented the data preprocessing pipeline for the HELP, MED, Semantic Fragments datasets. The first, second, and last two authors contributed to the writing of the paper. The third author provided constructive suggestions.

[2] The contribution of this part of research corresponds to the publication Feng et al. (2022). The original idea for the proposed model and its key components were formed by me, the second author, and my co-supervisors, i.e., the third and last author. The third author, Xiaodan Zhu, is the principal supervisor of this research. The implementation and code are my own contribution, except that the second author, Xiaoyu Yang, contributed to developing the sentence chunking module, surveying and comparing practical sentence alignment solutions, selecting the best-suited sentence encoder for the model, and helping present results of the experiments. All authors contributed to the writing of the paper.

Extending neuro-symbolic reasoning to multi-hop question answering (QA). We also explore how to extend the proposed model and the interpretation method to the applications of multi-hop QA. We proposed a model that accurately locates chains of useful evidence, which can be trained without direct supervision. We also build a neuro-symbolic QA model that performs natural-logic style reasoning on the chain of evidence. The proposed model achieves state-of-the-art performance in detecting medicine-to-medicine interactions from the medical text. [3]

[3] The contribution of this part of research corresponds to our publication Feng et al. (2021). The original idea and key components were formed by me, the second author, Mo Yu, and the fourth author, Xiaoxiao Guo. The implementation and code are my own contribution, except that the Recurrent Neural Network component was implemented by the second author. All authors provided constructive suggestions and contributed to the writing of the paper.

The following papers have been published in the course of research of this thesis:

Yufei Feng, Xiaoyu Yang, Michael Greenspan, Xiaodan Zhu. 2022. **Neuro-symbolic Natural Logic with Introspective Revision for Natural Language Inference.** *Transactions of the Association for Computational Linguistics (TACL)*, Volume 10, 2022.

Yufei Feng, Mo Yu, Wenhan Xiong, Xiaoxiao Guo, Junjie Huang, Shiyu Chang, Murray Campbell, Michael Greenspan, Xiaodan Zhu. 2019. **Learning to Recover Reasoning Chains for Multi-hop Question Answering via Cooperative Games.** *The 34th Canadian Conference on Arti cial Intelligence (Canadian AI 2021)*, Online.

Yufei Feng, Michael Greenspan, Xiaodan Zhu. 2021. **Enhancing Pretrained Models with Domain Knowledge.** *The 34th Canadian Conference on Arti cial Intelligence (Canadian AI 2021)*, Online.

Yufei Feng, Zi'ou Zheng, Quan Liu, Michael Greenspan, Xiaodan Zhu. 2020. **Exploring End-to-End Differentiable Natural Logic Modeling.** *The 28th International Conference on Computational Linguistics (COLING 2020)*, Online.

1.4 Organization of Thesis

The remainder of this thesis is organized as follows.

In Chapter 2, we provide background knowledge for neuro-symbolic methods, natural language inference (NLI) and multi-hop question answering (QA). We also explain in detail how natural logic formalism solves the NLI problem.

In Chapter 3, we present an end-to-end di erentiable natural logic framework, which leverages natural logic operations to solve NLI problems. We then discuss the limitations of the developed model. This chapter is based on our publication Feng et al. (2020).

In Chapter 4, we introduce an advanced neuro-symbolic natural logic reasoning system, which overcomes most of the predecessor's limitations. We also present Introspective Revision, a general algorithm that can incorporate commonsense knowledge base information into neuro-symbolic models. This chapter is based on our publication Feng et al. (2022).

Chapter 5 extends our research for multi-hop QA, where we develop neuro-symbolic models for the task of discovering symbolic reasoning chains in a weakly supervised setting, and the task of medical domain multi-hop QA. This chapter is based on our publication Feng et al. (2021).

Chapter 6 concludes the thesis.

Chapter 2

Background

2.1 Natural Language Inference

The semantic concepts of entailment and contradiction are central to natural language reasoning (Katz, 1972; Van Benthem et al., 2008; Bowman et al., 2015). Modeling these relations is essential to many natural language processing applications, which include but are not limited to information retrieval, commonsense reasoning, and image captioning. The natural language inference (NLI) task formally studies the relation between a premise sentence P and a hypothesis sentence H. According to the most popular de nition used by the mainstream NLI datasets (Bowman et al., 2015; Williams et al., 2018), the premise P entails the hypothesis H if H is true given P, i.e.,

$$P \text{ ñ } H \tag{2.1}$$

The premise P contradicts the hypothesis H if the hypothesis H is false given the premise P, i.e.,

$$P \text{ ñ } \quad H \tag{2.2}$$

In other cases, the premise *P* and the hypothesis *H* have a neutral relation. The NLI task is de ned as a three-way classi cation problem (Bowman et al., 2015; Williams et al., 2018), where a pair of sentences will be classi ed into one of the entailment, contradiction, or neutral relations. Some datasets merge contradiction and neutral relations as the non-entailment relation (Marelli et al., 2014; Yanaka et al., 2019a,b).

Some examples can be used to illustrate the NLI relations. The *entailment* relation usually happens in cases where the hypothesis is a paraphrase of the premise, for example, *Tom is walking outside* entails *Tom is taking a walk outdoors*; in cases where the hypothesis uses a broader concept than the premise, for example, *dog* entails *animal*; and in cases where the hypothesis contains fewer details compared with the premise, for example, *Tom is taking a walk outdoors* entails *Tom is outside.*

Contradiction happens where two statements cannot happen at the same time. For example, *Tom is taking a walk outdoors* contradicts *Tom is sleeping.* In some cases, there can be a disagreement between contradiction and neutral. For example, intuitively it is hard to determine whether the premise *A ship sinks in the Paci c* and the hypothesis *A ship sinks in the Atlantic* have a relationship of contradiction or neutral. In Bowman et al. (2015), this pair is labeled as a contradiction because these two events cannot happen in a single photo, even though they can co-occur with a minimal chance in general.

Neutral includes cases where entailment and contradiction do not hold. Reverse entailment also belongs to the *neutral* relation, where the hypothesis strictly entails the premise, for example, the premise *Tom is walking* and the hypothesis *Tom is walking outside in the park* have the relation *neutral*.

Though popular neural methods, such as ne-tuning pretrained transformers (Devlin et al., 2019a; Liu et al., 2019), t current NLI datasets well, it is still challenging to design models with faithful explainability and that generalize well to various linguistic and logical rules, such as downward monotonicity (Yanaka et al., 2019a,b) and systemacity (Yanaka

et al., 2020). In this thesis, instead of targeting the new state-of-the-art NLI accuracy, we design neuro-symbolic methods based on natural logic theory, which simultaneously achieves competitive performance on both standard benchmarks, such as Bowman et al. (2015), and various linguistic probes (Yanaka et al., 2019a,b, 2020). Meanwhile, our method provides faithful explanations for its inner mechanism.

Datasets that focus on di erent aspects of the NLI problems are proposed in the literature. Cooper et al. (1996) proposed FraCaS, an NLI dataset with 346 cases that include various linguistic phenomenons such monotonicity and conjunction. Bentivogli et al. (2009) proposed the RTE dataset, which includes around 6k NLI samples, and the label is not evenly distributed. The SICK dataset (Marelli et al., 2014) contains around 10k sentence pairs, some of which are designed to test negation and quanti cation. The Stanford Natural Language Inference (SNLI) dataset (Bowman et al., 2015) is a large annotated corpus (500k) for NLI, wherein the premises are inherited from Flicker30k image captions and the hypotheses are written by people on the Mechanical Turk. The MultiNLI dataset (Williams et al., 2017) is designed for NLI in the multiple-genre text. Adversarial NLI (Nie et al., 2019) is a more challenging dataset that constructs complex NLI samples in a human-in-the-loop manner. ContractNLI (Koreeda and Manning, 2021) are NLI applications on contracts reviewing, and Chen et al. (2019b) proposed a dataset to study the uncertainty in NLI labeling.

The aforementioned datasets may contain defects. Some analyses show that the SNLI dataset contains bias (Tan et al., 2019). For example, the negation and quanti er *every*, *all* are likely signs of contradiction. Moreover, a hypothesis-only model could achieve high accuracy on SNLI. Many probing datasets are proposed to investigate whether a model can learn certain meanings and rules from the data. Hossain et al. (2020) shows that negation is under-represented in the current NLI training and proposed a dataset to evaluate negation in NLI. Monotonicity, which is a linguistic concept that will be detailed later, is investigated in

the FraCaS, HELP (Yanaka et al., 2019b), and MED (Yanaka et al., 2019a). The analyses show that current datasets contain mostly samples written in upward monotonicity, and current models trained on SNLI have bad performance when generalizing to downward monotone cases (Yanaka et al., 2019b,a, 2020). ConjNLI (Saha et al., 2020) is proposed to investigate whether current models can handle conjunction. Some researchers focus on the model explanation for NLI. The e-SNLI dataset (Camburu et al., 2018) collects human-written explanation, and Nie et al. (2020) investigated the annotation agreement of the NLI explanation.

Multiple models have been proposed to solve the NLI problem. Current research papers use two major categories of natural language models. One category is the ESIM family (Chen et al., 2016, 2017a), where the premise and the hypothesis are encoded by recurrent neural networks and communicate through cross-attention. The other category is the pretrained transformers that includes but are not limited to Bert (Devlin et al., 2019a) and Roberta (Liu et al., 2019). Pretrained transformer networks achieved state-of-the-art performance on the SNLI and MNLI datasets, however, they are black-box models, and it has been shown that those models are prone to fit the dataset bias (Poliak et al., 2018; Hossain et al., 2020) and have limited ability to generalize to data with special linguistic phenomenons like downward monotonicity (Yanaka et al., 2019b,a, 2020; Geiger et al., 2020). Sinha et al. (2020) found that, in contrast to human beings, who struggle with ungrammatical sentences, current state-of-the-art NLI models work well even with permuted input, in which the input words are randomly shuffled.

Building an NLI model with interpretability is a goal with profound impact. Unlike many text classification applications such as sentiment analysis, which usually requires reasoning over one sentence, the NLI problem requires models to compare two sentences and to reason over the difference. As a result, interpreting a NLI model often involves reasoning over alignment (MacCartney et al., 2008; Jiang et al., 2021). To interpret an NLI model,

it is still possible to apply neural network explanation models for single-sentence applications (Lei et al., 2016; Ribeiro et al., 2016; Sundararajan et al., 2017; Chang et al., 2020; DeYoung et al., 2020), but many unique techniques for cross-sentence reasoning have also been proposed recently. Parikh et al. (2016) and Chen et al. (2017b) obtained model explanations from the attention weights in the co-attention layers; however, multiple researchers have argued that for many reasons (Serrano and Smith, 2019; DeYoung et al., 2020; Jain et al., 2020) the attention weight is not a good explanation as expected. To achieve better attention-based interpretability, Jiang et al. (2021) proposed to use an additional objective function on the cross-sentence attention model, which was shown to increase the sparsity and the faithfulness of the co-attention interpretation.

Another stream of work is focused on generating natural language explanations for given NLI samples. Those models adopt supervised training: the authors first obtain natural language explanations through crowdsourcing (Camburu et al., 2018), then train a generative model to fit the human-written explanations. In addition to the e-SNLI (Camburu et al., 2018) datasets, multiple models (Kumar and Talukdar, 2020; Chen et al., 2021a) are proposed to improve the quality of the generated explanations.

2.2 Multi-Hop Question Answering

While a large proportion of this thesis is focussed on NLI, we also propose neuro-symbolic models for multi-hop QA. The QA problem aims to answer a given text question Q by referring to a supportive document $D = \{P_1; P_2; ...; P_n\}$, where P_i are different pieces of text (i.e., sentences or paragraphs) in the document. The desired answer Ans can be either a word, a phrase, or a sentence depending on the question Q.

For the multi-hop QA task, to answer the question Q, a model needs to acquire at least two pieces of information (passages) from the document D. The QA task is closely related to NLI: it is possible to regard the document D as the premise and the question-answer pair

r*Q; Ans* as the hypothesis.

In contrast to the single-hop QA task, where the answer can be inferred from a single piece of supportive text (Rajpurkar et al., 2016; Seo et al., 2016; Devlin et al., 2019b), multi-hop QA requires the model to reason over multiple paragraphs or documents. Popular multi-hop QA challenges include Wikihop & Medhop (Welbl et al., 2018), HotpotQA (Yang et al., 2018), MultiRC (Chen et al., 2019a) and ComplexWebQuestions (Talmor and Berant, 2018).

To solve the multi-hop QA problem, most existing works use an evidence retriever to find the paragraphs that contain the correct answer and the supportive evidence (Asai et al., 2019; Chen et al., 2019a; Yichen Jiang and Bansal, 2019; Xiao et al., 2019; Tu et al., 2020). A reading comprehension model is then applied over the retrieved paragraphs to obtain the correct answer (Min et al., 2019b; Nishida et al., 2019). Yichen Jiang and Bansal (2019) proposed to dynamically construct an evidence graph for better reading comprehension, and Fang et al. (2020) proposed a hierarchical graph neural network for evidence representation.

There is recent interest in predicting reasoning chains for multi-hop QA (Asai et al., 2019; Chen et al., 2019a; Ding et al., 2019). These researchers all adopted a supervised setting, in which a large amount of annotated chains are available. However, it is more practical to assume that the reasoning chains for multi-hop QA are not available during training. From this perspective, we highlight Wang et al. (2018b); Min et al. (2019a) and Perez et al. (2019), who tried to train the QA model from noisy weakly-supervised evidence.

2.3 Neuro-Symbolic Methods

Neural Theorem Proving (NTP) Rocktaschel and Riedel (2017) proposed a Neural Theorem Prover framework, which combines theorem-proving algorithms such as Prolog (Gallaire and Minker, 1978) with a neural network. The core of NTP is constructing a

tree of proof trajectories T for a given goal statement and judging each trajectory to validate the goal statement. Each proof trajectory, starting from the root node (original goal) in T, is a chain of rules that can potentially prove the goal. For example, given an original goal proof: GrandpaOf(ABE, BART), a possible trajectory in the tree is FatherOf(ABE, HOMER) ^ ParentOf(HOMER, BART). This trajectory mimics the backward chaining algorithm, where two sub-goals are chained together to prove the original goal.

Minervini et al. (2018) improved the NTP by making the construction of search tree T more e cient. They proposed a heuristic algorithm that only expands T with rules fetched by the nearest neighbor search. The follow-up work (Minervini et al., 2020) ultilized an additional *select* module to propose relevant rules, thus avoiding searching all the candidate rule decompositions over the database. The idea of NTP was extended by Weber et al. (2019) to deal with multi-hop QA.

Latent Programs and Neural Module Network (NMN) To answer a complex question, NMN de nes several task-speci c neural modules. Each module is designed to achieve a typical goal such as *nding the dog in the picture* or *getting the dog's color*. Andreas et al. (2016a) leveraged a NMN to solve visual QA tasks. The model rst parses the question to a program, then assembles a prediction network according to the program. The whole model is trained end-to-end using the image-answer pairs in the training set. In this work, the program parsing step is a deterministic process achieved by a rule-based parser.

In Mao et al. (2019), both the program generator and the executor involve a neural network, so the generator and the executor are jointly trained using the REINFORCE algorithm (Williams, 1992). They proposed to use curriculum learning to facilitate joint training. Speci cally, they presented the model with easy samples rst so the REINFORCE algorithm could nd positive rewards relatively more easily. After the parser obtained some

ability to generate meaningful programs, they ne-tuned the whole model on more complex samples in the dataset. They also found that without a designed curriculum, the model fails to train.

Gupta et al. (2019a) extended the NMN to reasoning over text. Unlike the original NMN for visual QA, the text module network is designed to answer natural language questions. The model rst uses LSTM or BERT to transform the question to a program, which is a list of connected sub-modules. Then the inference network is assembled according to the required sub-modules. Because it is not possible to parse complex questions using the formal method (this is unlike the case in Andreas et al. (2016a)), the text module network jointly trains the neural parser and the inference model in a weakly supervised manner. In practice, the model applies an auxiliary training objective as a regularizer to learn the distribution of the latent program. Moreover, Gupta et al. (2019b) annotated the ground truth logic program for 10% of the training data to facilitate the parser training. In a parallel work, Chen et al. (2019d) designed a similar model that rst proposes a program and then executes it for nal prediction. To successfully generate the latent program, they obtained plausible programs for part of the training samples using an exhaustive search, and they used the programs in the searching results that produced correct answers as the direct supervision. Both works are evaluated on the DROP dataset (Dua et al., 2019), where symbolic operators, including counting, sorting, and simple mathematical operations, are required to answer the question.

Compared to high-performance black-box models such as QDGAT (Chen et al., 2020a), neural symbolic models provide a human interpretable clue about how they make decisions. To obtain the rationales and debug the model, one can investigate each neural module's input and execution results. However, current neural symbolic models have inferior performance compared to black-box neural models because there are hard problems in the dataset that the human-designed task-speci c programs could not easily handle. Meanwhile, it is

hard to jointly learn a program generator and the parameters of the program modules simultaneously.

Probabilistic Logic In contrast to neural symbolic methods (Andreas et al., 2016a; Gupta et al., 2019a; Mao et al., 2019) that parse the question into executable programs with sub-modules, it is possible to parse the question into a probabilistic program: di erent modules operate over the same probability space, and the program yields a probability distribution as output. The Deep ProbLog (Manhaeve et al., 2018) model uses a formal method to parse questions into probabilistic programs and then uses maximum likelihood method to train the model parameters. Amizadeh et al. (2020b) converted rst-order logic expressions to probabilistic programs and performed analytical experiments on visual QA.

Advances in Optimizing Neural-Symbolic Models Guu et al. (2017) discussed the scenario where the output y is generated by a latent process z, and the goal is to jointly learn the distribution of z and y conditioned on the input X. Their paper shows that the REINFORCE objective and the maximum marginal likelihood (MML) are closely related. The paper also shows that both the REINFORCE and MML methods su er from the spurious problem: incorrect programs execute to the correct answer. Moreover, the paper also shows that programs that obtain positive rewards have a high likelihood of being sampled, no matter whether correct or spurious, will be strengthened so that the rich get richer and the poor get poorer. Guu et al. (2017) also showed evidence that shorter programs will be favored due to the sampling process. To alleviate these problems, they proposed a randomized beam search algorithm to address the spurious problem and a meritocratic update rule to adjust the learning rate for rarely sampled programs.

Li et al. (2020) proposed a backward search method to make the REINFORCE exploration more e cient. They found that if generating the correct latent program is challenging, the policy gradient is likely to be zero because no program is lucky enough to hit the correct

output during training. They proposed a back search algorithm that searches for potential corrections for an erroneous program that is close to the correct results. During training, the back search algorithm maintains a priority queue. Starting from the final program output, the priority queue pops the operation that is most likely to be wrong and uses a solver to guess the possible correction that leads to the positive reward. Once a modified program is found, it is used as a target program to supervise the program generation network. They showed that this learning by correction framework is equivalent to sampling programs from the posterior distribution of the latent program z given the final label y. By doing the back searching, the model avoids sampling zero-reward programs and obtains training rewards more efficiently. They also showed that combined with a random walk search, their model is equivalent to a Metropolis-Hasting sampler on the posterior distribution.

2.4 Natural Logic

Rather than performing deduction over an abstract logical form, natural logic (Lakoff, 1970; van Benthem, 1988; Valencia, 1991; Van Benthem, 1995; Nairn et al., 2006; MacCartney, 2009; MacCartney and Manning, 2009; Icard, 2012; Angeli and Manning, 2014) models logical inferences in natural language by operating directly on the structure of language. Natural logic allows for a wide range of intuitive inferences in a conceptually clean way (MacCartney, 2009; Angeli and Manning, 2014) and hence provides a good framework for developing explainable neural NLI models. Specifically, our work is motivated by the natural logic variant proposed by MacCartney and Manning (2009), for which we will provide more background as follows.

Natural Logic Relations The natural logic inference system proposed by MacCartney and Manning (2009) operates by mutating spans of text in a premise to obtain the

Relation	Relation Name	Example
$x \equiv y$	equivalence	*mom* ≡ *mother*
$x \sqsubset y$	forward entailment	*cat* ⊏ *animal*
$x \sqsupset y$	reverse entailment	*animal* ⊐ *cat*
$x \wedge y$	negation	*human* ∧ *nonhuman*
$x \mid y$	alternation	*cat* | *dog*
$x \smile y$	cover	*animal* ⌣ *nonhuman*
$x \# y$	independence	*happy* # *student*

Table 2.1: Seven natural logic relations in MacCartney and Manning (2009).

Quantifier & Connective	Proj.	Input Relation r					
		⊏	⊐	∧	|	⌣	#
all	$^{arg1}\pi_{r,q}$	⊐	⊏	|	#	|	#
	$^{arg2}\pi_{r,q}$	⊏	⊐	|	|	#	#
some	$^{arg1}\pi_{r,q}$	⊏	⊐	⌣	#	⌣	#
	$^{arg2}\pi_{r,q}$	⊏	⊐	⌣	#	⌣	#
not	$\pi_{r,q}$	⊐	⊏	∧	⌣	|	#

Table 2.2: The projection table for monotonicity reasoning.

corresponding hypothesis sentence and generates proofs based on the natural logic relations of the mutations. To extend the entailment relations to consider semantic exclusion, MacCartney and Manning (2009) introduced seven set-theoretic relations B for modeling entailment relations between two spans of texts (see Table 2.1 for some examples).

Assuming the availability of the alignment between a premise and hypothesis, the system first infers the relations between aligned pairs of words or phrases. Consider the following example:

The child does not love sports.

The kid doesn't like table-tennis.

The relation between *the child* and *the kid* is *equivalence* ( ), which is the same as the relation between *does not love* and *doesn't like*, while *sports* reversely entails (⊐) *table-tennis*.

'		€	•	^	|	`	#
		€	•	^	|	`	#
€	€	€	#	|	|	#	#
•	•	#	•	`	#	`	#
^	^	`	|		•	€	#
|	|	#	|	€	#	€	#
`	`	`	#	•	•	#	#
#	#	#	#	#	#	#	#

Table 2.3: Results (Icard, 2012) of composing one relation (row) with another relation (column).

Monotonicity Reasoning The next step is monotonicity inference. Monotonicity is a pervasive feature of natural language that explains the impact of semantic composition on entailment relations (Van Benthem, 1986; Valencia, 1991; Icard and Moss, 2014). Similar to the monotone functions in calculus, upward monotone keeps the entailment relation when the argument \increases" (e.g., *cat* € *animal*). Downward monotone keeps the entailment relation when the argument \decreases" (e.g., in *all animals* € *all cats*). The system performs monotonicity inference through a projection function : B Ñ B, which is determined by the context and projection rules. Table 2.2 (MacCartney, 2009; Angeli and Manning, 2014) shows some examples. Consider the last row in the table | it shows how the project function works in the negated context following the negation word *not*. Speci cally, this row shows seven relations that p*r*q will output, given the corresponding input relations *r*. For example, if the input relation is *forward entailment* (€), the function projects it to *reverse entailment* (•); i.e., p`€'q `•'. As a result, in the example shown above, the *reverse entailment* relation (•) between *sports* and *table-tennis* will be projected to *forward entailment* (€) in the negated context.

Relation Composition Building on that, the system aggregates/composes the projected local relations to obtain the inferential relation between a premise and hypothesis. Speci cally, Table 2.3 ((MacCartney, 2009; MacCartney and Manning, 2009; Angeli and

Manning, 2014)) shows the composition function when a relation (in a row) is composed with another (in a column). In practice, multiple compositions as such are performed in sequential order or from leaves to the root along a constituency parse tree. MacCartney (2009) shows that di erent orders of compositions yield consistent results except in some rare arti cial cases. Therefore, many works, including ours here, perform a sequential (left-to-right) composition. In the example shown above, composing two *equivalence* ( ) with *forward entailment* (€) yields *forward entailment* (€), resulting in a prediction that the premise entails the hypothesis.

Recent works (Hu et al., 2020; Kalouli et al., 2020; Chen et al., 2021b) have started to combine neural networks with logic-based components, however, the works mentioned above did not approach the NLI problem from the neuro-symbolic perspective. Compared to Hu et al. (2020) and Chen et al. (2021b), the neuro-symbolic model we propose in this thesis is more exible at learning from large annotated datasets, and is more robust towards language variations and noise.

Logic Form vs. Surface Form Unlike some popular neural symbolic models (Mao et al., 2019; Amizadeh et al., 2020a; Khot et al., 2020), which design domain-speci c language for the dataset of interest, we intend to explore neuro-symbolic models that only rely on language surface form (MacCartney, 2009). For example, to verify the statement *there is a red bottle to the right of a cat*, Mao et al. (2019) generated (by performing semantic parsing) and executed the nested program

IS_EQUAL(*red*, GET_COLOR(FILTER(*bottle*, RELATE(*right*, FILTER(*cat*))))),

which requires the model to rst nd the cat from all objects, then to select all objects to the right of the cat, then to nd the bottle, then to get the color of the bottle, then to compare the color with red.

Instead of using nested logic programs, surface form reasoning does not involve semantic

parsing, and it only focuses on important components of the statement, such as *red bottle*, *a cat*, and verifies the relationship between each component and the supportive evidence, such as the provided image. Though reasoning over logic forms is often mathematically sound, it is difficult to generalize, because as the statement becomes increasingly complex or flexible, a designed domain-specific language or logic system may fail to handle some of the cases. As a result, the error from the semantic parsing system may increase (Subramanian et al., 2020). Reasoning over surface form is more tolerant to data variations: a complex statement can be decomposed (often based on the syntax of the statement) into multiple important components, and the model performs reasoning by first verifying each components then aggregating the verification results.

In this thesis, we explore how to design models that leverage surface form reasoning to solve the NLI problem and how to extend the idea to more challenging applications of open-domain multi-hop QA (Rajpurkar et al., 2016; Welbl et al., 2018; Yang et al., 2018).

2.5 Transformer Network and Pretrained Language Models

Some of the models proposed in this thesis, together with several important baseline models, are based on the pretrained transformer network (Vaswani et al., 2017; Devlin et al., 2019b; Liu et al., 2019; Radford et al., 2019). In this section, we give a brief introduction of the transformer network, and we refer readers to Vaswani et al. (2017) for a detailed implementation.

Transformer network (Vaswani et al., 2017), a pure attention-based network, is designed to accelerate the model training and inference on sequential data. Compared to the LSTM network, which contains a loop along the time axis and is hard to be computed fully in parallel, the transformer network uses repeated attention layers to encode the input data: information of different input units (e.g., words) is processed in parallel and the sequence position information is treated as features.

As the central part of the transformer layer, a multi-head attention layer transforms each input token representation and combines information in the sequence with a weighted sum. For a single attention head, let

$$H \quad \mathrm{r}h_1; h_2; :::h_n\mathrm{s} \tag{2.3}$$

denote the sequence of input embedding, and let three independent linear projections Q (query), K (key), and V (value) be computed through feed-forward layers. The attention weight is the normalized dot-product of Q and K.

$$Q \quad W_q \quad H \tag{2.4}$$

$$K \quad W_k \quad H \tag{2.5}$$

$$V \quad W_v \quad H \tag{2.6}$$

$$H^1 \quad softmax\mathrm{p}\frac{Q}{?}\frac{K^T}{\overline{d_k}}\mathrm{q}V \tag{2.7}$$

The attention output H^1 is a weighted sum of representations V, and d_k in the normalizing term is the sequence length of K.

The multi-head architecture greatly increases the information that can be encoded in the network. In a single layer with n heads, the transformer network has n independent $K; Q; V$ transformations. Representation vectors from di erent heads are concatenated together before they go through two fully connected layers. A skip connection, and a batch normalization layer, are applied after each attention layer, and after the fully connected layer in each transformer layer.

To encode the positional information (e.g., the order of the words in the input) of the sequence, Vaswani et al. (2017) proposed to add the positional encoding to the token representation. The positional encoding is a set of trainable vectors, one for each position

in the sentence, and is initialized with sine/cosine functions.

Pretrained Transformers To achieve good performance on a speci c natural language processing task, a transformer network goes through a pretraining process and a ne-tuning process. The pretraining process usually contains a language modeling objective, while di erent pretrained models have their unique auxiliary objectives. The GPT (Radford et al., 2018, 2019; Brown et al., 2020a) style of pretraining requires the transformer network to predict the next word based on the part of the text that appeared before it. The transformer network is trained as a left-to-right language model that maximizes the next word likelihood

$$J(\theta) = \sum_{t=2}^{N} P(w_t | w_{1:t-1}; \theta), \tag{2.8}$$

where the parameter θ is the transformer network weights and w_t is the word at time t.

Unlike GPT (Radford et al., 2018), the language model BERT (Devlin et al., 2019b) is trained in a bidirectional manner. The model is required to predict masked tokens in the sentence, and the language model training objective can be formulated as

$$J(\theta) = \sum_{t \in C}^{N} P(w_t | w_{\setminus C}; \theta) \tag{2.9}$$

where C is a set of the masked tokens (similar to the cloze test were some words in the sentence are missing and waiting to be lled). BERT also uses an additional next-sentence prediction objective. We refer readers to Devlin et al. (2019b) for details on the masked token prediction task and the next-sentence prediction objective.

Recent works also proposed multi-modal pretraining. For example, Dosovitskiy et al. (2020) pretrained a transformer network with image and text. Similar works include Chen et al. (2020b), Su et al. (2019), Gan et al. (2020) and Radford et al. (2021), which are proven to deliver advanced performance on visual-QA-related tasks.

Fine-tuning Transformers Pretrained transformer models can be easily fine-tuned on specific downstream tasks: we can concatenate relevant information in the input and feed it to the transformer encoder. The output (vector) representation of the transformer can be used for prediction.

Take BERT as an example, the input pieces ABC and DE, can be fed to BERT as follows:

[CLS] A B C [SEP] D E [SEP]

where [CLS] and [SEP] are special tokens that indicate the start and the separation respectively.

For the NLI task, ABC can be the premise sentence and DE can be the hypothesis. The final output vector of the token [CLS] can be used to generate a three-way classification prediction.

For the QA task, ABC can be the question and DE can be the supportive evidence paragraphs. The final output vector can be used to generate the final answer, or it is also possible to predict a span in the text as the answer.

2.6 Interpreting Neural Network Models

The past few years have seen a significant growth in the field of explainable models, due to the widespread applications of high-accuracy deep learning models and end-to-end training, which often lack interpretability.

According to the categorization of Wallace et al. (2020), some researchers (including but are not limited to Adi et al. (2016); Shi et al. (2016); Conneau et al. (2018)) developed probing classifiers to analyze the relationship between internal representations of the black-box models and different linguistic properties. Other researchers (including but are not limited to McCoy et al. (2019); Gardner et al. (2020); Ribeiro et al. (2020)) analyzed

model behaviors with systematically designed challenging datasets or diagnostic samples.

Feature Influence. One major stream of model explainability research is to study the influences of the input features, and methods in this stream are often model-agnostic, that is, the explanation method can be applied to any model of interest, as long as the dataset input format is compatible with the explanation model, or the model is open to gradient analysis.

Many models in this stream explain the black-box models by generating a saliency map, which shows the relative importance of a specific feature towards the desired output, for example, the model may assign a high saliency score on *bad* when explaining a negative sentiment prediction. Among saliency map models, some approaches (Simonyan et al., 2013; Smilkov et al., 2017; Shrikumar et al., 2017; Sundararajan et al., 2017; Han et al., 2020) produced the saliency map by analyzing the gradient of the model prediction. There are also similar approaches (Bach et al., 2015; Shrikumar et al., 2017; Selvaraju et al., 2017) designed for computer vision applications.

Instead of backpropagating the gradient to the input, other models produce a saliency map by perturbing the input. Li et al. (2016) and Feng et al. (2018) explained text classification models by erasing words in the input sentences. Ribeiro et al. (2016) proposed the LIME system, which uses a linear classifier to approximate the feature importance. Lundberg and Lee (2017) proposed to measure the contribution of each feature by calculating its SHAP value. Because removing some input words may dramatically change the original meaning of the input and lead to non-sensical inputs, apart from removing words or phrases, some researchers (Ribeiro et al., 2018; Ebrahimi et al., 2017; Iyyer et al., 2018) also considered using adversarial perturbations to analyze neural models.

Attention-Based Explanation Attention heatmap has long been considered as an explanation method for neural network models. Chen et al. (2017b) and Parikh et al. (2016)

leveraged cross-sentence attention to explain NLI reasoning, Jiang et al. (2021) proposed to use multiple objectives to improve attention explanation, while many researchers (Jain et al., 2020; Serrano and Smith, 2019; DeYoung et al., 2020) argued that attention heatmap cannot serve as a reliable explanation method.

Sample-Based Explanation Another stream of researchers try to explain model behaviors with important samples. Given a speci c image, Yeh et al. (2018) explained image classi cation models by searching for representative samples in the training data that induce the test prediction. Koh and Liang (2017) proposed the data in uence function to measure each training sample's in uence on a speci c test case.

Other Methods Parallel to the aforementioned model explanation methods, Lei et al. (2016) proposed to extract rationales by training a rationale extraction model and a prediction model simultaneously. The method was further improved by (Chang et al., 2020) and Yu et al. (2021) to produce more reliable rationales. Some authors generate natural language sentences as explanations (Camburu et al., 2018; Kumar and Talukdar, 2020; Chen et al., 2021a; Situ et al., 2021); however, Jacovi and Goldberg (2020) argued that natural language explanations generated from those models may not be faithful to the prediction model.

Chapter 3

Exploring End-to-End Differentiable Natural Logic Models

3.1 Introduction

As mentioned in Chapter 2.4, the good performances of end-to-end deep neural networks are often achieved by tting large labeled datasets, and despite the exibility in the architectures, the neural networks are black-box models that provide few explanations for their inner mechanism. As a result, it is hard to judge whether a well-trained neural network has successfully learned to perform the desired reasoning or whether it simply learned to produce correct answers by leveraging super cial patterns or dataset biases (Clark et al., 2019; Tan et al., 2019; Clark et al., 2020; DeYoung et al., 2020; Hossain et al., 2020). Recent works show that state-of-the-art deep natural language inference (NLI) models, though achieving accuracies that are close to human performance, still failed to capture many aspects of human reasoning, which include but are not limited to interpretability (Lei et al., 2016; DeYoung et al., 2020), monotonicity inference (Yanaka et al., 2019a,b; Geiger et al., 2020), and systematic compositionality (Yanaka et al., 2020).

To tackle the aforementioned challenges, one popular solution is to further test and retrain deep neural network models on complementary datasets, which are specially designed

to re ect certain linguistic phenomena. These datasets can also serve as extra stress tests to expose potential weaknesses of the models. Examples of these datasets can be found in Yanaka et al. (2019a,b, 2020) and Geiger et al. (2020). However, this solution has limitations: while the neural network learns to memorize additional patterns and improves its performance on the stress tests, the model remains a black box and its inner mechanism is still uncontrollable. As a result, the model may generalize poorly to unseen samples beyond the complementary datasets, even if the unseen samples and the training samples share similar underlying logic.

In this chapter, we explore a di erent approach | a di erentiable NLI framework that combines natural logic symbolic formalism (Lako , 1970; van Benthem, 1988; Valencia, 1991; Van Benthem, 1995; Nairn et al., 2006; MacCartney, 2009; MacCartney and Manning, 2009; Icard, 2012; Angeli and Manning, 2014) with a neural network. At the lower level, neural network models are used to encode the input data and to produce intermediate predictions such as relations. Taking advantage of the distributed representation, the neural network model learns to automatically extract and encode robust features by back-propagation. At the higher level, the framework reasons by performing symbolic operations. The symbolic operators ensure that the model makes decisions following a controlled symbolic reasoning method, and the operations performed can also serve as a faithful explanation for the model's inner mechanism. The whole framework is end-to-end di erentiable.

To build such a neuro-symbolic NLI model, we need to face and at least partially solve the following challenges:

- **How to align text components of the premise and hypothesis?** A core part of natural logic is to infer the relations between the aligned text components; however, existing tools for text alignment cannot be easily transferred to NLI for several reasons (MacCartney et al., 2008), among which we highlight that (1) current text alignment tools are developed for machine translation or text paraphrasing, while

NLI focuses on text entailment and contradiction, and (2) few annotated alignment resources for monolingual data are available. Given that an ideal text alignment tool for NLI is not available, one potential solution for alignment is to leverage the soft-attention network (Xu et al., 2015; Chen et al., 2017b; Vaswani et al., 2017) to bypass explicit alignment; however, this approach potentially harms the interpretability (DeYoung et al., 2020).

- **How to formulate the natural logic as a probabilistic model?** As is shown in Section 2.4, natural logic provides a formal method to solve NLI. In most real-life cases, the proof method operates with human effort and cannot easily be automated as a statistical model. Thus, before designing the neuro-symbolic natural logic model, we need to re-formulate natural logic as a probabilistic model and define the likelihood function, which will be optimized by back-propagation and gradient descent.

- **How to evaluate NLI models from the angle of natural logic?** Though accuracy is widely used as the benchmark for NLI, we need to design metrics to evaluate models' interpretability and to what extent the neural model reasons following natural logic.

In the rest of this chapter, we propose solutions for the aforementioned challenges. In section 3.2, we first explain how the natural logic reasoning process (mentioned in Section 2.4) can be formulated as a probabilistic state-transition model, and we then describe the details of the end-to-end differentiable natural logic reasoning model. The experiment setups are presented in Section 3.3, and the evaluation results, together with the analysis, are shown in Section 3.4.

3.2 End-to-End Differentiable Natural Logic Model

Natural logic is a self-contained logic framework for NLI (MacCartney, 2009; MacCartney and Manning, 2009). However, directly applying the natural logic mentioned in Section 2.4 to samples in SNLI is challenging. For example, according to MacCartney (2009), as the first step of the reasoning, sentence components from the premise and the hypothesis need to be aligned, and one natural logic relation needs to be inferred from each aligned text pair. In MacCartney's natural logic formalism, the alignment is performed by human beings, while in machine learning applications, aligning the premise and the hypothesis pairs is especially challenging (MacCartney et al., 2008). Moreover, it is an open question how many aligned pairs should be acquired from the premise and the hypothesis. For example, given the premise *The dog is sleeping during the morning.* and the hypothesis *The dog is sleeping during the night*, it makes sense to align *during the morning* with *during the night*, while according to MacCartney's natural logic it is also possible to treat *during* v.s. *during*, *the* v.s. *the*, and *morning* v.s. *night* as three aligned text pairs. To facilitate the development of the end-to-end natural logic framework, in this chapter we make minor modifications to MacCartney's natural logic, and re-formulate the natural logic reasoning process as state transitions within the probability space, the minimum reasoning unit of which being a word.

3.2.1 Formulating Natural Logic Reasoning as State Transitions

To determine the number of aligned text pairs in the natural logic reasoning process, we use the hypothesis tokens as anchors: for each hypothesis token x_j^h at j^{th} position j 1; :::; $|H|$, we align x_j^h with text X_j^p in the premise, where X_j^p is a collection (possibly being empty) of tokens in the premise that potentially associate with x_j^h. Then we treat the pair x$x_j^h; X_j^p$y as an aligned text pair. In this way, we resolved the ambiguity in the number of the aligned text pairs, and the number of the aligned pairs is equivalent to the number of hypothesis

tokens in the input. Again taking the pair of premise and hypothesis above as the example, *during* v.s. *during*, *the* v.s. *the*, and *morning* v.s. *night* are three aligned text pairs, and the framework acquires $|H|$ 7 aligned text pairs ($|H|$ is the length of the hypothesis) before proceeding to further reasoning steps. In addition to the reduction of ambiguity, this hypothesis-anchored formulation partially simpli es the alignment problem. Given the hypothesis tokens, there is no need to explicitly compute X_j^p, which, as explained before, is an intimidating challenge.

Given a sequence of aligned pairs x$x_j^h; X_j^p$y; j 1; :::; $|H|$ and their corresponding natural logic relations r_j, we formulate the reasoning process as state transitions within seven natural logic relations (or states) listed in Table 2.3. Starting with the initial default state z_0 ` 'p*equivalence*q, for each step j, we compose the current state z_j with the incoming relation $r_{j\ 1}$, which results in a new state $z_{j\ 1}$ according to the composition rules in Table 2.1. After consuming all the input relations, the nal state $z_{|H|}$ indicates the nal reasoning outcome, which can be translated to the NLI label predictions. Speci cally, for the example in Figure 3.1, we have ground truth relations r_1 r_3 r_4 ` ', $r_2$ `€', r_5 `|'. According to Table 2.3, the corresponding state trajectory $z$ is: $z_1$ ` ', z_2 z_3 z_4 `€', and $z_5$ `|'. The nal state of the trajectory z determines the nal reasoning outcome, which leads to a *contradiction* label. Note that here z_i P B where B is the set of seven relations listed in Table 2.1.

Although the proposed formulation simulates how humans compare text components of the premise and the hypothesis according to the natural logic theory, the formulation is not unique. MacCartney (2009) found that natural logic theory is self-contained in that, except in some rare and arti cial cases, the nal outcome remains unchanged regardless of the order in which the aligned text pair relations are composed. In this chapter, we focus on the cases where the relations r are composed in a left-to-right order or bottom-up along the constituency tree.

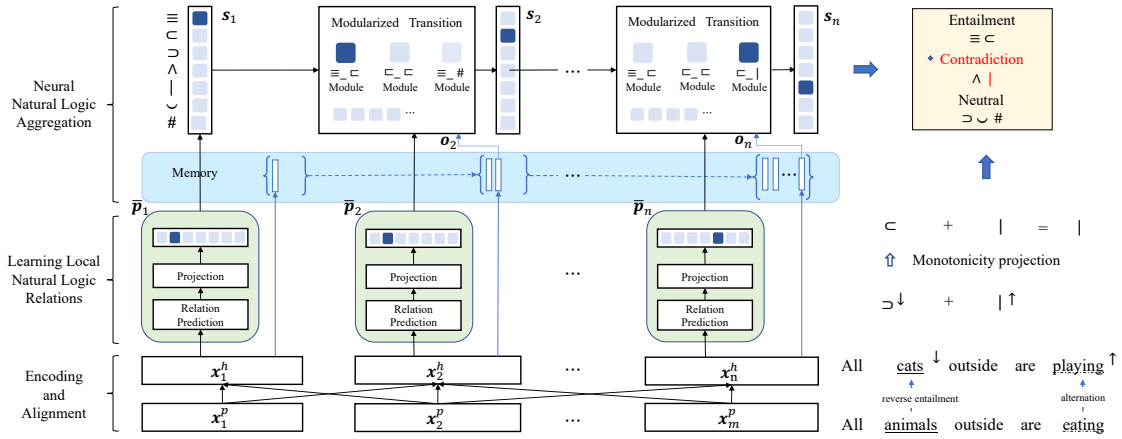

Figure 3.1: A high-level view of the proposed neural natural logic model.

To build a di erentiable model that executes according to the proposed formulation, we need to depict the proposed state transitions in the probability space. At each step j, r_j, and z_j are random variables with a sample space B. Then we have:

$$p(z_j = k \mid h, p) = \sum_{s \in B} p(z_j = s \mid h, p, z_{j-1} = k) \, p(r_j = s \mid h, p, z_{j-1} = k) \tag{3.1}$$

where $k \in B$. And the goal of the training is to maximize the likelihood function of the correct NLI label y:

$$p(y \mid X) = \sum_{z \in Z} p(y \mid z, h, p) \, p(z \mid X), \tag{3.2}$$

where $X = (X^p, X^h)$ is the input premise and hypothesis, and Z is the collection of all possible z trajectories. In the following sections, we present details of how $p(y \mid X)$ is computed and optimized over labeled training data.

3.2.2 Encoding and Alignment

Recent research has shown the e ectiveness of distributed representations for encoding lexicons and their semantic relations. We use word embedding and neural networks to learn lexical representations to capture natural logic related semantics. Let X^p be a premise sentence and X^h the corresponding hypothesis sentence,

$$X^p \quad \text{t } x_1^p; x_2^p; :::; x_m^p \text{ u} \tag{3.3}$$

$$X^h \quad \text{t } x_1^h; x_2^h; :::; x_n^h \text{ u}; \tag{3.4}$$

where m and n are the number of word tokens in the premise and hypothesis, repspectively. Each sentence is fed into a multi-layer BiLSTM, for which

$$a_i \quad BiLSTM \text{ p} X^p; i \text{q} \tag{3.5}$$

denotes the i^{th} hidden vector at the top layer of the BiLSTM, encoding the i^{th} token and its context in the premise. Similarly, we use

$$b_j \quad BiLSTM \text{ p} X^h; j \text{q} \tag{3.6}$$

to denote the hidden vector at the j^{th} position at the top layer of the BiLSTM that encodes the hypothesis.

In this thesis, we focus on understanding neural natural logic itself, without being further confounded by di erent ways of exploring knowledge external to the training data, e.g., via pretraining.

Many models can be used to capture cross-sentence attention. Focusing on the training data, the approach proposed in Chen et al. (2017b) has been widely used in the NLI literature as a baseline. We follow the work to compute cross-sentence attention weight

$e_{ij} = a_i^T b_j$ for each pair $\langle a_i, b_j \rangle$. Specifically, for each b_j in the hypothesis, the corresponding content in a premise is weighted summed as

$$\tilde{b}_j = \sum_{i=1}^{m} \frac{\exp(e_{ij})}{\sum_{k=1}^{m} \exp(e_{kj})} a_i; \tag{3.7}$$

which will be used together with b_j to learn local lexical-level inference relations (refer to Chen et al. (2017b) for more details).

In addition, we compute a *hard alignment indicator* $_j$, and $_j = 1$ *if and only if* $x_i^p = x_j^h$, where $i = argmax_{i \in \{1,\ldots,m\}} e'_{ij}$.[1] That is, for each word token x_j^h in the hypothesis, we record the token x_i^p in the premise that has the maximum attention value e'_{ij}. If the word token x_i^p and x_j^h are the same word type, we let $_j = 1$, which will be used to help reduce the search space in aggregation.

3.2.3 Learning Local Natural Logic Relation

Given a sequence of alignment $\{\langle \tilde{b}_1, b_1 \rangle, \ldots, \langle \tilde{b}_j, b_j \rangle, \ldots, \langle \tilde{b}_n, b_n \rangle\}$, we use a bi-linear model to compute each pair's probabilistic distribution p_j over the natural logic relations $\mathcal{B}$:

$$p_j = softmax(f_s(\tilde{b}_j, b_j)) = softmax(\tilde{b}_j^T M^T b_j)$$

In the scoring function f_s, each type of relation $k \in \mathcal{B}$ has its own weight matrix $M_k \in \mathbb{R}^{d \times d}$, which is a slice of the tensor $M \in \mathbb{R}^{d \times d \times |\mathcal{B}|}$, where d is the dimensionality of b_j or $\tilde{b}_j$. We use *softmax* to normalize the values to be a distribution over $\mathcal{B}$. Among several alternatives we used, the bi-linear model achieves the best performance on the development dataset, and we use it in our final framework.

[1] Here e'_{ij} is the cross-attention weight obtained from the ESIM model (Chen et al., 2017b) trained on SNLI.

3.2.4 Local Relation Constraints

Same as in many other weakly supervised setups, we do not have direct supervision signals here to learn logic relationships at the lexical level; instead, the supervision signals are backpropagated from the overall sentence-level NLI errors. To reduce the search space and alleviate the *spurious* problem (Guu et al., 2017) in which incorrect local inference relations and their aggregation produce correct sentence-level NLI labels,[2] we adopt several strategies as follows.

Symmetric Inference Parameter Sharing: We make the *forward entailment* (€) and *reverse entailment* (•) relations share the same parameters. Speci cally, to compute $p_j^{\bullet}$, we reverse the order of $\langle \tilde{b}_j, b_j \rangle$ to reuse $M_{€}^{T}$ in the following scoring function, where $M_{€}^{T}$ is a matrix in M^T that corresponds to the *forward entailment* (€) relation.

$$f_s^{\bullet}(\tilde{b}_j, b_j) \quad f_s^{€}(b_j, \tilde{b}_j) \quad b_j^T M_{€}^{T} \tilde{b}_j \tag{3.8}$$

Equivalence Constraint: A token pair will be assigned the *equivalence* relation ( ), if $_j$ learned above in the alignment stage takes the value of 1:

$$\text{if } _j \quad 1; \text{ we let } p_j \quad 1 \tag{3.9}$$

Collapse Constraints: We suppress the relations *negation* (^) and *cover* (`):

$$p_j^{\wedge} \quad 0; \quad p_j^{`} \quad 0 \tag{3.10}$$

Inspired by Angeli and Manning (2014), we suppress the *negation* relation (^) because

[2] In an extreme case, if a model predicts the rst aligned word pair between a premise and hypothesis to be a relationship that is consistent with the ground-truth NLI label at the sentence level, the model can choose to ignore all other pairs that follow, and make the correct sentence-level prediction by using the rst pair prediction only, even if the aggregation sequence z is incorrect.

its behavior is almost same as that of *alternation* (|) in natural logic aggregation, as shown in Table 2.3, avoiding the co-linearity problem when training on datasets without double negation samples. We also suppress the *cover* relation (`) because it is extremely rare in current natural language inference datasets.

3.2.5 Projected Distribution

With the predicted seven-dimensional probability vector p_j being ready, our model uses a projection operator to re-organize the distribution according to the projectivity of the corresponding input hypothesis word at position j.

Unlike the discrete \hard" projection used in the conventional natural logic, e.g., projecting the rst argument of *all* from *reverse entailment* to *forward entailment*, we apply \soft" projection over relation probability distribution p_j. Speci cally, based on the projection Table 2.2, we convert the original probability distribution p_j to the projected distribution $\bar{p}_j$:

$$p_j^{k^1} \quad \sum_k p_j^k \mathbb{1}(p \ pkq \quad k^1q; \tag{3.11}$$

where $\mathbb{1}(p\ q)$ is the indicator function, k is the original relation, and k^1 is the projected relation. Consider the pair of sentences in Figure 3.1 and suppose the pair *eating* vs. *playing* have a probability of 0.8 to be *alternation* (|) and 0.1 to be *negation* (^). According to the projectivity of the second argument of the quanti er *all* in Table 2.2, both relations are projected to alternation (|): $^{playing}p|q \quad ^{playing}p\hat{}q \quad |$.

So after projection, $p_5^| \quad p_5^| \quad p_5^\hat{} \quad 0{:}9$, where the subscript 5 is the index of the word token *playing* in the hypothesis.

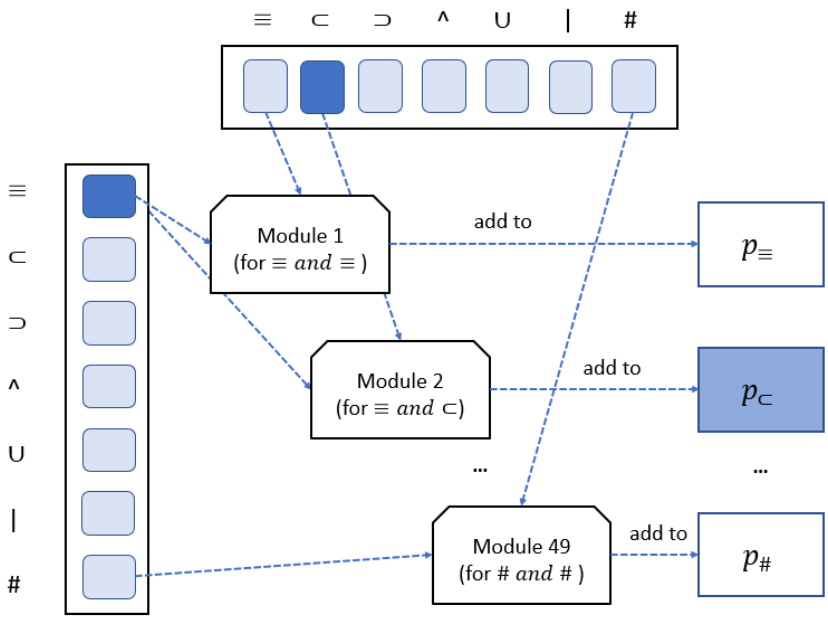

Figure 3.2: The module network for natural logic aggregation.

3.2.6 Aggregation

We propose to leverage the module networks (Andreas et al., 2016b; Gupta et al., 2019b) to perform neural natural logic aggregation, which is enhanced by a memory network component to leverage the powerful ability in modeling the context. Figure 3.2 shows the proposed neural natural logic aggregation network.

Speci cally, at each time step j, our aggregation algorithm computes a distribution $p(z_j|X) = softmax(s_j)$, where $s_j = \{s_j^k\}$ is a set of logits. s_j^k is the one corresponding to $p(z_j = k|X)$ for relation $k \in \mathcal{B}$. Our model computes s_j^k with Equation 3.12.

$$
\begin{aligned}
s_j^k \quad & \sum_{u \in B} \sum_{v \in B} G^{u \cdot v}(s_{j-1}^u, p_j^v; o_j) \mathbb{1}(u \cdot v \equiv k) \\
& \sum_{u \in B} \sum_{v \in B} (s_{j-1}^u - p_j^v - g^{u \cdot v}(o_j)) \mathbb{1}(u \cdot v \equiv k) \\
& \sum_{u \in B} s_{j-1}^u \sum_{v \in B} p_j^v - g^{u \cdot v}(o_j) \mathbb{1}(u \cdot v \equiv k);
\end{aligned} \tag{3.12}
$$

At time step 1, s_1 is initialized with p_1. At any other time step $t > 1$, we invoke modules $G^{u \cdot v}(\cdot)$ to derive s_j. Specifically, in our network each relation aggregation in Table 2.3, i.e., $u \cdot v = (u, v \in B)$, has its own module $G^{u \cdot v}(\cdot)$. Now, given the previous s_{j-1} and the current *projected local relation* distribution p_j, s_j can be computed by marginalizing the Cartesian product $s_{j-1} \bar{\times} p_j^T$ according to aggregation Table 2.3. More specifically, we first compute the Cartesian product $s_{j-1} \bar{\times} p_j^T$, which is weighted by the memory $g^{u \cdot v}(o_j)$. Then for all modules with output being the same relation k according to Table 2.3, the modules' output are summed up, where $\mathbb{1}(\cdot)$ is the indicator function.

Below we discuss how the memory network response o_j is calculated. In this thesis, we propose a memory network component (Weston et al., 2014; Sukhbaatar et al., 2015) to enhance our module aggregation network, aiming to better model contextual information. Specifically, at time step j, we store memory vectors $\{m_1, \ldots, m_j\}$ and the corresponding output vectors $\{c_1, \ldots, c_j\}$ in the memory. The query vector q_j scans the memory and computes the match between itself and memory vectors by taking the inner product followed

by a *softmax*:

$$q_j = f_q(\tilde{b}_j, b_j) \tag{3.13}$$

$$m_j = f_m(\tilde{b}_j, b_j) \tag{3.14}$$

$$c_j = f_c(\tilde{b}_j, b_j) \tag{3.15}$$

$$\alpha_{j,t} = softmax(q_j^T m_t), \quad t = 1, \ldots, j \tag{3.16}$$

The query, memory, and output vectors are functions of aligned token representation $(\tilde{b}_j, b_j)$, typically modeled by two feed-forward layers.

The response vector o_j is computed by the weighted sum over stored outputs vectors c_j and is used in the module network discussed above:

$$o_j = \sum_{t=1}^{j} \alpha_{j,t} c_t \tag{3.17}$$

where o_j encodes all historical transitions and their context and is then incorporated into Equation 3.12.

In addition to the sequential aggregation we discuss above in which we perform aggregation left-to-right over a premise and hypothesis pair, we also perform the aggregation on the binarized constituency parses, where the aggregation is performed on a tree structure. For node j in the constituency tree, we define a random variable z_j which represents the reasoning states upon seeing the node j and sub-tree, and we use s_j to denote the distribution of z_j. We initialize s_j with projected relation distribution p_j if node j is the leaf node. Iteratively, the distribution s_j for each non-leaf node is computed by aggregating its left child (lc) and right child (rc):

$$s_j^k = \sum_{u \in B} \sum_{v \in B} G^{u \circ v}(s_{lc}^u, s_{rc}^v, o_j) \mathbb{1}(u \circ v = k) \tag{3.18}$$

where o_j is the memory network response vector which is computed on the information of all nodes that have already been visited.

3.2.7 Objective Function

The final prediction of sentence relation is computed with the distribution of hidden state s_n at the last time step (or the root node if reasoning is performed over the constituency tree). We follow the work of Angeli and Manning (2014) and group relation *equivalence* ( ) and *forward entailment* (€) to be *entailment*; *negation* (^) and *alternation* (|) to be *contradiction*, and; *reverse entailment* (•), *cover* (`) and *independent* (#) to be *neutral*. We apply a variant of hard-EM training method (Min et al., 2019a), which selects the most likely relation: $p_{entailment}$ $\max(s_n; s_n^{€})$, $p_{contradiction}$ $\max(s_n^{\wedge}; s_n^{|})$, and $p_{neutral}$ $\max(s_n^{•}; s_n^{`}; s_n^{\#})$. After applying softmax, we obtain the prediction probability, which can be used to compute the cross-entropy loss.

3.3 Experiments Setup

We evaluate the proposed differentiable natural logic model with multiple experiments. First, we test the model's NLI performance on multiple stress test datasets, each of which evaluates whether the model can generalize to samples with specific linguistic properties (e.g., downward monotonicity). Then we construct a new test dataset NaturalLogic-2Hop to evaluate the interpretability of the proposed differentiable natural logic model.

3.3.1 Stress Test Datasets

We use three datasets designed for studying monotonicity-based reasoning, i.e., HELP (Yanaka et al., 2019b), MED (Yanaka et al., 2019a), and the monotonicity subset of Semantic Fragments (Richardson et al., 2020). The models are trained on SNLI (Bowman et al., 2015) and evaluated on the stress test datasets.

The HELP dataset has 35,891 inference pairs, which are automatically generated by conducting lexical substitution or deletion on one sentence to obtain the other, given natural logic polarity information of each word token and syntactic structure of sentences. The MED dataset contains 5,382 human-generated inference pairs by either asking crowd workers to perform the generation or manually collecting the pairs from linguistics publications. The monotonicity subset of Semantic Fragments is automatically generated with a controlled set of rules and lexicons, which contains around 2,000 pairs. Because the pairs with the contradiction relation in the Semantic Fragments dataset are obtained by changing quanti ers, which are out of the scope of the natural logic formalism that we use, we do not include this subset in our experiments.

3.3.2 NaturalLogic-2Hop Dataset

We create a new *2-hop* dataset to evaluate how the inner mechanism of the proposed model aligns with natural logic reasoning performed by human beings. Existing datasets (Yanaka et al., 2019b,a) lack ground-truth labels for evaluating natural logic aggregation at each time step, and most of their samples are characterized by the *1-hop* aggregation, where a premise and the corresponding hypothesis di er only by one span of text. As a result, we need to develop new test benchmarks for natural-logic-based models. In our *2-hop* dataset, the premise and hypothesis di er by two edits of word/phrase insertion, deletion, or substitution. Our dataset also provides ground-truth aggregation output $f z_1, ..., z_j, ... z_n g$ to help assess models' performance on natural logic reasoning.

Figure 3.3 shows an example of the 2-hop dataset. The premise and hypothesis di er by two edits of word/phrase insertion, deletion, or substitution, which are highlighted in di erent colors. The dataset provides the ground truth aggregation results at each time step (the *equivalence* relation is the default relation in the beginning and is hence not included) and the word locations/indices associated with each edit. The 2-hop dataset is developed

with the following three steps:

Premise: Some delegates finished the survey on time.
Hypothesis: Some individuals finished the survey.
Label: Entailment

Edit 1: delegates → individuals **Type**: hypernym **Relation**: forward_entailment
Location: premise: {1} hypothesis: {1}

Edit 2: on time → [ ] **Type**: delete **Relation**: forward_entailment
Location: premise: {5, 6} hypothesis: { }

Ground truth of aggregation:
Position {1}: forward entailment
Position {4}: forward entailment

Figure 3.3: An example of the 2-hop dataset.

Identifying MED Relations: Because most sentence pairs in the MED dataset only di er by one word/phrase edit, it is straightforward to determine the location of the insertion, deletion, or replacement. For insertion and deletion, we follow Angeli and Manning (2014) and treat the relation as *reverse entailment* (•) and *forward entailment* (€), respectively. We set aside the replacement samples because we cannot determine their relations without human labeling. To ensure the identi ed natural logic relations are correct, we compare the labels provided in MED with labels determined by MacCartney's natural logic theory and remove samples in which the labels do not agree, yielding roughly 1,1000 sentence pairs.

Adding One More Hop of Relations: We ask human annotators to help replace a noun either in the premise or the hypothesis with another word, and the relation between the substituted and substituting word is one of t ; €; •; |; #u. The rst two authors of Feng et al. (2020) are the annotators for the word replacement. We require that the candidate

words to be replaced are not children or parents of any previously identi ed di erences over parsing trees. To facilitate the dataset creation process, we rst use WordNet to automatically suggest substituting words (e.g., hypernyms or hyponyms). This procedure yields roughly 28,000 candidate sentence pairs. Then we remove some types of word replacement if either of the annotators thinks the replacement leads to unnatural sentences. This replacement operation yields 5,858 sentence pairs, and the premise and the hypothesis of each example now di er by two edits.

Determining Labels: We apply projection operation and natural logic aggregation according to MacCartney and Manning (2009) to determine the three-way NLI labels for the generated 2-hop sentence pairs. We also record the ground-truth relations of each hop of aggregation output.

We manually assess the data quality on 300 sentence pairs (100 for each category). We nd that on average 3% of the samples have either incorrect labels or wrong intermediate aggregation output (4% in category *entailment*, 4% in category *neutral*, and 1% in *contradiction*). Those mistakes are mainly produced by incorrect parser-identi ed polarity. The 2-hop dataset can represent real-life sentences because the samples are generated by modifying samples in the MED dataset, where the sentences are manually generated or collected.[3]

3.3.3 Evaluation Metrics for Interpretability

We evaluate the intermediate aggregations of the proposed model with precision, recall, and F1 score. Precision is the number of correctly performed aggregations, divided by the total number of aggregations performed by a model. The recall is the number of correctly performed aggregations, divided by the total number of aggregations presented in

[3]We release the dataset following the ethical agreement of the original dataset MED (Yanaka et al., 2019a), and we do not impose ethical conditions on our part of annotation.

Model	Monotonicity Fragments (%)	HELP (%)	MED (%)	Natural Logic 2-Hop (%)
ESIM	66.18	55.27	51.78	45.13
BERT-base	50.58	51.40	45.88	49.33
Neural Nat. Log. (seq.)	66.03	58.23	**52.47**	**60.14**
Neural Nat. Log. (tree)	**66.47**	**63.95**	47.57	59.97

Table 3.1: Test accuracy of the models.

the ground-truth annotation. Note that we only consider aggregations at time step t when $\hat{z}_t \neq \hat{z}_{t-1}$. Because by default the starting state $\hat{z}_0$ ` ', so if $\hat{z}_1$ ` ', we do not count this degenerate case.

3.3.4 Implementation Details:

Following Chen et al. (2017b), hidden vectors in our model are 300 dimensional. We use pretrained 300-dimensional 840B GloVe vectors (Pennington et al., 2014b) to initialize our word embeddings. All word embeddings are trainable after being initialized. We apply a dropout rate of $p = 0.5$. Adam (Kingma and Ba, 2015) is used as our optimizer, and the first momentum is set to be 0.9 and the second 0.999. The batch size is set to 32, and the initial learning rate is 0.0004. We train ESIM and our neural natural logic models for 32 epochs and use the development set to select models for testing. We use default hyper-parameters specified in Devlin et al. (2019b) and train the BERT-base model for 3 epochs.

3.4 Experiment Results

3.4.1 Performance on Stress Tests

Table 3.1 shows the test accuracy of different models on the four datasets that are designed specifically for evaluating monotonicity-based inference. Following Richardson et al. (2020) and Yanaka et al. (2019a), we train the models on SNLI (Bowman et al., 2015) and test

Model	**HELP Dev** (%) Up Mono.	**HELP Test** (%) Down Mono.
ESIM	95.25	21.49
BERT-base	98.63	13.71
Neural Nat. Log. (seq.)	91.20	63.08
Neural Nat. Log. (tree)	90.62	**70.80**

Table 3.2: Test accuracy of the models (continued).

on these di erent test sets. The proposed models, in general, achieve better performances on these four datasets than ESIM (Chen et al., 2017b) and BERT (Devlin et al., 2019b). The di erence is more prominent in the 2-hop dataset, which requires the system to have a better aggregation ability to make the nal prediction.

To demonstrate how the models generalize between the upward and downward monotone, we train the models with HELP's upward monotone subset and test on the downward monotone subset. A system that can better model monotonicity should achieve more robust performance. Speci cally, we split the upward monotone subset of the HELP dataset into the training set (6k training examples) and the development set (1:5k examples). We train all models on the training split and select models with the highest development accuracy. We test all models on the HELP downward monotone subset (21k examples). Table 3.2 shows that although ESIM and BERT achieve very high development accuracy on the upward data, they fail to generalize to the downward monotone test set. The proposed models generalize well and achieve better test accuracy on the downward monotone datasets.

	Model	Precision	Recall	F1
(1)	**Neural Nat. Log. (seq.)**	**0.54**	**0.49**	**0.51**
(2)	(1) w/o. **Memory / Module**	0.49	0.46	0.47
(3)	(2) w/o. **Local Rel. Constraints**	0.12	0.15	0.13

Table 3.3: Evaluation of models' aggregation performance on the 2-hop dataset.

3.4.2 Interpretability Evaluation

The proposed model provides inference explainability by accessing natural logic's aggregation and decision paths. We visualize the predicted local relation distribution and the aggregation with a heatmap. We also design experiments to measure the model's interpretability in a quantitative manner, based on the 2-hop dataset we have developed. Figure 3.4 shows an example of the 2-hop dataset, together with the visualization of the intermediate aggregation decisions. From left to right, the first sub figure shows the cross-sentence attention between the premise (x-axis) and hypothesis (y-axis), where a darker color corresponds to a larger attention weight. In the second sub figure, for each word in the hypothesis (y-axis), the predicted distribution of lexical-level logical relations is shown along the x-axis. The third sub figure shows the aggregation output. For example, on the second row, the aggregation has already been performed over the first two words b_1 = *the*" and b_2 = *animals*" using their lexical relation distributions, which have been shown in the second sub figure and are, in turn, computed from the first sub figure using x$\tilde{b}_1; b_1$y and x$\tilde{b}_2; b_2$y. Given ` ' ' `€' = `€', we can see that on the second row, a large probability mass has been put on € (i.e., *ent_f* in the figure).

We further perform a quantitative analysis on the aggregation performance. We analyze the sequential aggregation. Specifically, for the 2-hop dataset in which we have access to the aggregation decisions: $\hat{z}$ = t$\hat{z}_1; \hat{z}_2; :::; \hat{z}_n$u, where $\hat{z}_j$ is the aggregation result at time step j, we evaluate the models by comparing the estimated $\hat{z}$ with the ground truth z. We evaluate the intermediate aggregation steps of the proposed model with precision, recall, and

Premise: Two dogs, the gray poodle high in the air, play on the grass.
Hypothesis: The animals are lying on the bed.
Label: Contradiction

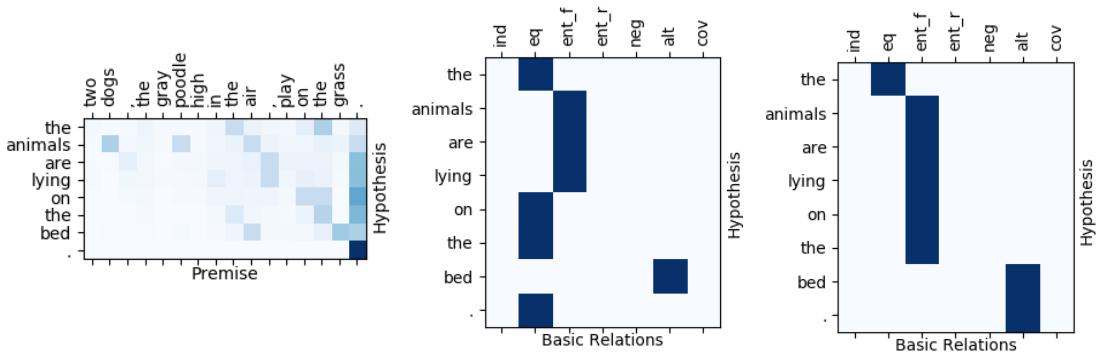

Figure 3.4: A visualized example for natural logic explanation.

F1 score. Precision is the number of correctly performed aggregations, divided by the total number of aggregations performed by a model. Recall is the number of correctly performed aggregations, divided by the total number of aggregations presented in the ground-truth annotation. Note that we only consider aggregations at time step t when $\hat{z}_t \neq \hat{z}_{t-1}$. Since by default the starting state $\hat{z}_0$ = ` ', so if $\hat{z}_1$ = ` ', we do not count this degenerate case.

Table 3.3 shows the results. Because ESIM and BERT do not produce intermediate aggregation results, they are not included in the table. The ablation analysis shows that both the memory/module component and the local relation constraints help the model to learn intermediate natural logic aggregation. We can also see that further work is desirable to improve the performance on aggregation prediction because there is still much room to improve modeling performance on this. As part of our e orts, we have also performed component training to leverage WordNet (Miller, 1998) and ConceptNet (Liu and Singh, 2004) to help determine lexical relations. This approach is not particularly e ective because

the lexical pairs from these knowledge bases only cover a small percentage of pairs that need to be modeled.

3.5 Conclusions and Future Works

3.5.1 Summary of Contribution

In this chapter, we proposed an end-to-end differentiable natural logic framework for NLI. To the best of our knowledge, this is the first neuro-symbolic framework in the literature that explores how to combine the strength of the neural network and natural logic to solve the task of NLI.

In this work, we reformulated the natural logic proof system developed by MacCartney and Manning (2009) to a probabilistic model and implemented the reformulated model with an end-to-end neural network.

During the model development, we are the first to observe the `*entangling*' problem (please refer to a detailed formal explanation of the problem in Section 4). We proposed multiple lexical constraints to mitigate the entangling problem, which makes our proposed model able to produce reasonable model explanations.

We design multiple evaluations for end-to-end natural logic models, which include a monotonicity-related generalization test and a new challenge test dataset for natural logic explanation evaluation. Based on the evaluation, we show that our model is capable of handling the linguistic phenomenon of monotonicity reasoning, and provides reasonable proof for the prediction.

3.5.2 Problems Not Yet Fully Resolved

The proposed differentiable natural logic model achieved promising results on NLI, with a special performance gain on the stress tests and interpretability. However, some experiments

expose the weakness of the current model, leading to potential improvements.

Alignment and the Local Relation Constraints. Due to the difficulty of explicitly aligning text components in the premise and the hypothesis, the proposed model uses attention as soft alignment, while applying strong lexical constraints to prevent the loss of interpretability. Table 3.3 shows the constraints are essential to the model interpretability. Imposing strong constraints may have negative effects on models' performance, and in this case, it is hard to measure how much interpretability is coming from the model reasoning or how much is simply a result of predefined constraints. As a result, is worthwhile to explore how to make the model learn the alignment freely and independently.

The Use of Commonsense Knowledge Bases The current end-to-end natural logic model cannot leverage external knowledge bases such as WordNet (Miller, 1998) and ConceptNet, which are crucial to the proposed model because they provide hints to the local relation modeling. Some works (Glockner et al., 2018; Talmor et al., 2020) show that pre-trained language models like BERT (Devlin et al., 2019a) and Roberta (Liu et al., 2019) can implicitly capture lexical knowledge, including hypernym relations. However, the knowledge is not learned and presented systematically. Even if some researchers explore ways to use knowledge base by concatenating the input with knowledge base embedding (Bordes et al., 2013), the knowledge base embedding cannot explicitly affect the intermediate reasoning process of the end-to-end differentiable logic model. It is worthwhile to investigate how commonsense knowledge bases can be utilized during training.

In the next chapter, we propose an advanced model whose interpretability does not rely on local relation constraints. The new model also has a robust mechanism to integrate commonsense knowledge bases into natural logic reasoning.

Chapter 4

Neuro-Symbolic Natural Logic with Introspective Revision

4.1 Introduction

In the previous chapter, we introduced an end-to-end differentiable natural logic model for natural language inference (NLI), and as mentioned in the discussions, the proposed model has several limitations, detailed as follows:

Problem from the Lexical Constraints One of the most intriguing points of the discussions is that, in the end-to-end differentiable natural logic model, the model's interpretability strongly relies on lexical constraints. Especially, the equivalence constraint improves the interpretability, but it will inevitably cause trouble in some special cases.

Remember that the equivalence constraint forces two identical words in the premise and hypothesis to have relation *equivalence*( ), given the condition that the pair of identical words are also aligned by a separately trained ESIM (Chen et al., 2017b) model. In real applications, the premise and the hypothesis may share few common words, reducing the potential usefulness of the equivalence constraint. In addition, in the following downward monotonic example, the equivalence constraint may cause problems:

P: No dog is running on the eld.

H: No dog is running.

The premise does not entail the hypothesis, because though no dogs are running on the eld, there could be dogs running elsewhere. According to natural logic, `*running on the eld*' forward entails(•) `*running*', and the forward entailment (€) relation will be ipped to reverse entailment (•) due to the downward monotone quanti er `*no*', and the nal answer is *neutral*. However, according to the equivalence constraint, all words in the hypothesis should have a relation *equivalence*( ), resulting in a nal answer of *entailment*. From this example, we can see that rigid lexical constraints may lead to errors, especially in downward monotone cases and when the hypothesis is shorter than the premise.

In this chapter, we solve this problem by removing the lexical constraints, so the model is required to learn the alignment automatically.

Word-Level Relations Vs. Phrase-Level Relations The state transition scheme proposed in the previous chapter is slightly di erent from human reasoning. In Section 3.2, the local relations and the state transitions are de ned on the lexical level. The number of transition steps matches the number of words in the hypothesis. However, for a human, although there are no golden rules on how to chunk a given sentence, it makes sense to treat some words together as a phrase, e.g., *in the morning* or *take a trip*. In one natural logic reasoning example presented in MacCartney (2009), consecutive word tokens like *hesitate* and *to* are grouped as a single phrase *hesitate to*, which is treated as one unit during reasoning.

Grouping words into phrases (chunks of words) has profound bene ts. The existing NLI datasets provide no intermediate supervision for learning local relations during training, and our model needs to explore the action space of size 5^m to nd the path leading to the correct

nal NLI label, which is challenging if m (the number of aggregation steps) is large. As a result, it is desirable to properly reduce the reasoning steps and increase reasoning e ciency.

Incorporating Knowledge from External Resources As is discussed in Section 3.5.2, it is worthwhile to investigate how commonsense knowledge bases can be utilized during training because it provides direct hints for the local natural logic relations. For example, from the commonsense knowledge bases such as WordNet (Miller, 1998) and ConceptNet (Liu and Singh, 2004), we may nd the knowledge that *table tennis* is a kind of *sports*. This piece of knowledge is useful because it tells the model that the relation between *sports* and *table tennis* is reverse entailment ($\bullet$), or forward entailment ($\bullet$) vice versa, which can be used directly as intermediate supervision during model training.

However, the end-to-end di erentiable natural logic model leaves little space for utilizing such commonsense knowledge, given the objective function (Equation 3.12) does not explicitly consider the probability of each local relation. The model is trained with a maximal marginal likelihood (Guu et al., 2017) objective: at each step in the state transition, the probability of each local relation is summed up before being passed to the next step, and only the likelihood at the nal step is used in the loss function. To make full use of the external hints from the commonsense knowledge base, we need to design a model that leverages the probability of each local relation at each step in the objective function.

Moreover, it is challenging to design knowledge-aware NLI systems, considering the presence of noise and the ambiguous word meaning in the knowledge base, and even if the relation of a pair of words is present in the database, the relation may not be useful in determining the local relations, due to linguistic variations. In this chapter, we also discuss how to leverage noisy and ambiguous word relationships in the knowledge base to improve natural logic reasoning.

4.2 Neuro-symbolic Natural Logic with Introspective Revision

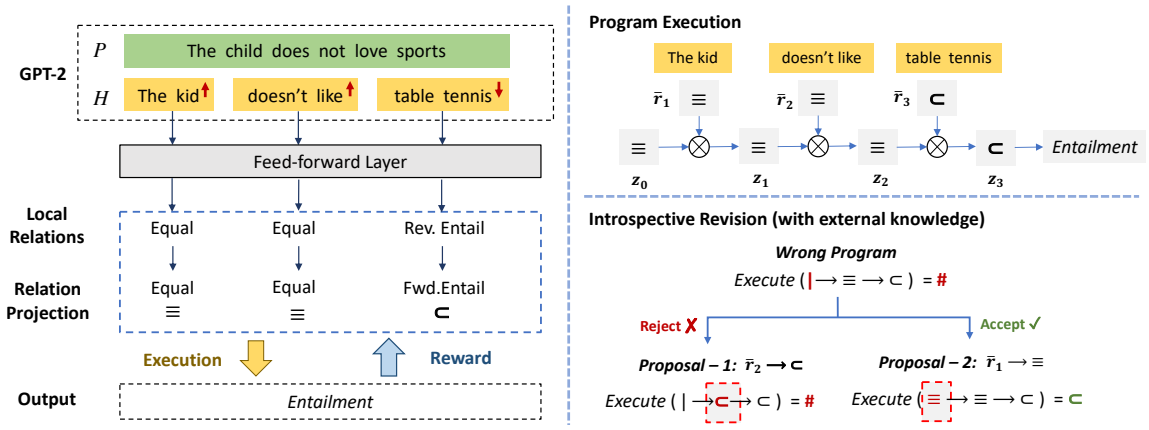

Figure 4.1: An overview of the proposed neuro-symbolic natural logic framework.

Motivated by the analysis of the limitations above, we propose an enhancement of the natural logic model. This section introduces our neural natural logic framework based on the proposed Reinforcement Learning with the Introspective Revision approach. We start with local relation modeling, in which caution needs to be taken to avoid the input entangling problem, which can seriously harm the model's interpretability. By viewing the local relation distribution as the stochastic policy, our model then samples and rewards speci c reasoning paths through a policy gradient, in which the Introspective Revision model can modify intermediate symbolic reasoning steps to discover better reward-earning operations and leverages external knowledge to alleviate spurious reasoning and training ine ciency.

4.2.1 Local Relation Modeling

We use phrases/chunks instead of words as the basic reasoning units. The primary motivation for chunking is to shorten the reasoning paths and hence reduce the number of possible paths, both of which make the reasoning process more e cient. Motivated by

Ouyang and McKeown (2019), we segment the premise P and the hypothesis H into several phrases/chunks. Specifically, we first extract noun phrases with spaCy (Honnibal et al., 2020) and then group the continuous spans of words between two noun phrases as chunks. As shown in Fig. 4.1, by identifying the noun phrases "*the kid*" and "*table tennis*", the hypothesis sentence H is segmented into three chunks. We denote the number of chunks in the hypothesis as m, and the t-th hypothesis chunk (and its vectorized representation) as s_t. Similarly the t'-th premise phrase is denoted as $\mathbf{s}_{t'}$.

As the first step of the neuro-symbolic natural logic, we use a neural network to model the local natural logic relation between each hypothesis phrase s_t and its associated premise constituents. However, accurately finding the hard alignment between s_t and the corresponding phrase $\mathbf{s}_{t'}$ in the premise is a hard problem (MacCartney et al., 2008). Current state-of-the-art NLI systems, like BERT (Devlin et al., 2019b), use bi-directional soft attention to model the cross-sentence relationship, however, we observe that it tends to fully *entangle* the input (DeYoung et al., 2020). Consider the top-left example in Fig. 4.1. If we use BERT to encode the input sentences, then the bi-directional attention model can infer the final NLI label solely based on the last-layer hidden states of the first hypothesis phrase "*the kid*" because the contextualized representation of this phrase *entangles* the information of the whole input through attention. Consequently, the hidden states of the phrase contain global information, thus not being suitable for modeling local relations.

To alleviate the undesired entangling, we model local relations with uni-directional attention (such as GPT-2). On the one hand, uni-directional attention prevents entangling future inputs. For example, in Fig. 4.1, the phrase "*table tennis*" will not affect the relation prediction anchored on "*The kid*". On the other hand, although the last hypothesis phrase attends to all previous inputs, without knowing whether the current phrase is the ending one (the future inputs are not available), the model cannot skip predicting the natural logic relation at the current phrase s_t and postpone all the required reasoning to the last phrase.

Speci cally, suppose a model always predicts *equivalence* ( ) at each step t and postpones its nal decision to the last hypothesis phrase. Without knowing that *table tennis*" is the ending phrase, the model can predict *equivalence* ( ) for \table tennis" and wait to make a better decision upon seeing the next input phrase, which actually does not exist. Failing to make timely local predictions that lead to the correct label before running out of the hypothesis phrases, the model will receive a negative reward in the end. In this way, the model is encouraged to be more careful in predicting the local relation for each hypothesis phrase. We also develop a model that achieves local relations by masking both the past and future hypothesis chunks. Compared to such a model, we will show later (Table 4.3) that the uni-directional attention model performs better, partly because it preserves the structure of the pretrained GPT-2 model.

Speci cally, we propose to model the local relation between s_t and the premise P, which can be e ciently achieved by the pretrained GPT-2 model (Radford et al., 2019). We concatenate a premise and hypothesis as the input and separate them with a special token xsepy. The contextualized encoding h for the -th hypothesis token is extracted from the GPT-2 last-layer hidden states at the corresponding location:

$$h \quad GPT\text{-}2\text{p}P; H_{1:\ \text{q}} \tag{4.1}$$

For the t-th phrase in the hypothesis s_t $H_{\ 1:\ 2}$, which starts from position $_1$ and ends at position $_2$, we concatenate features of the starting token $h_{\ 1}$ and the ending token $h_{\ 2}$ as the vectorized phrase representation:

$$s_t \quad Concat\text{p}h_{\ 1}; h_{\ 2}\text{q} \tag{4.2}$$

We use a feed-forward network f with ReLU activation to model the local natural logic relations between the hypothesis phrase s_t and its potential counterpart in the premise. The

feed-forward network outputs 7 logits that correspond to the seven natural logic relations listed in Table 2.1. The logits are converted with *softmax* to obtain the local relation distribution:

$$p_t \quad softmax(f(s_t)); \tag{4.3}$$

Intuitively, the model learns to align each hypothesis phrase s_t with the corresponding premise constituents through attention, and combines information from both sources to model local relations. In practice, the local relation distribution is de ned over ve relations: we merge relation *negation* (^) and *alternation* (|) because they have similar behaviors in Table 2.3, and we suppress *cover* (`), because it is rare in the current NLI datasets. Hence we only need to model ve natural logic relation types, following Feng et al. (2020).

4.2.2 Natural Logic Program

We propose to use reinforcement learning to develop neural natural logic, which views the local relation distribution p_t as the stochastic policy. At each time step t, the model samples a relation $r_t \in B$ according to the policy, and we treat the sequence of sampled relations $\{r_t\}_{t=1}^{m}$ as a symbolic program, which executes to produce the nal inferential relation between a premise and hypothesis. According to the best of our knowledge, this is the rst model that integrates reinforcement learning with natural logic.

Built on the natural logic formalism of MacCartney and Manning (2009), a projection function (Eq. 4.5) maps r_t to a new relation $\bar{r}_t$. In our model, the projection function is determined by the projectivity feature from the StanfordCoreNLP *natlog* parser[1]. For each input token, the projectivity feature speci es the projected relation $\bar{r}_t$ for each input relation r_t. In this work, we extend the token-level projectivity to handle phrases: for a phrase with multiple tokens, is determined by the projectivity of the rst token in the phrase. In Fig. 4.1, the projectivity of the phrase "*table tennis*" is determined by the

[1] https://stanfordnlp.github.io/CoreNLP/natlog.html

rst token *table*", and projects the predicted *reverse entailment* (•) relation to *forward entailment* (€).

$$r_t \quad sampling(p_t) \tag{4.4}$$

$$r_t \quad r_t \tag{4.5}$$

The program then composes the projected relations $\{r_t\}_{t=1}^{m}$ to derive the nal relation prediction, as shown in top-right part in Fig. 4.1. Speci cally, at time step $t = 0$, the executor starts with the default state z_0 *equivalence* ( ). For each hypothesis phrase $s_t; t \geq 0$, the program performs one step update to compose the previous state z_{t-1} with the projected relation r_t:

$$z_t \quad step(z_{t-1}; r_t) \tag{4.6}$$

The nal prediction is yielded from the last state z_m of program execution. Following Angeli and Manning (2014), we group *equivalence* ( ) and *forward entailment* (€) as ***entailment***; *negation* (^) and *alternation* (|) as ***contradiction***, and; *reverse entailment* (•), *cover* (Y), and *independence* (#) as ***neutral***.

Rewards and Optimization: During training, we reward the model when the program executes to the correct answer. Given a sequence of local relations $r = \{r_t\}_{t=1}^{m}$, at each step t the model receives a reward R_t as follows:

$$R_t = \begin{cases} \; ; & \text{if } Execute(r) = y \\ \; ^{m-t}; & \text{if } Execute(r) = y; \end{cases} \tag{4.7}$$

where is the constant reward unit, $\gamma \in (0; 1]$ is the discount factor, and y is the ground-truth label. In addition to Eq. 4.7, di erent rewards are applied under two exceptional cases: (1) if at step t there is no chance for the program to get a positive reward, then

the execution is terminated and the model receives an immediate reward R_t ; (2) when the true label is *entailment*, the model receives no positive reward if the last state z_m is *equivalence* ( ). In this way, we encourage the model to select at least one forward entailment (€) relation during prediction, instead of aggregating a sequence of *equivalence* ( ) for all entailment cases. In the current NLI datasets, it is less likely that the premise and hypothesis sentences are semantically equivalent to each other.

We apply the REINFORCE (Williams, 1992) algorithm to optimize the model parameters. During training, the local relations r_t are sampled from the predicted distribution, and we minimize the policy gradient objective:

$$J \quad \sum_{t=1}^{m} \log p_t(r_t) \quad R_t; \tag{4.8}$$

where $p_t(r_t)$ is the probability that corresponds to the sampled relation r_t. During the test, the model picks the relation with the largest probability.

Relation Augmentation: It can be hard to learn the *reverse entailment* ($\bullet$) relation from the existing NLI datasets because the relation of a pair of premise and hypothesis is labeled as *neutral*, if *H entails P* and *P does not entails H*. To help the model distinguish *reverse entailment* $p \bullet q$ from *independence* $p \# q$, both of which result in the NLI label *neutral*, we perform relation augmentation to create samples whose hypothesis entails the premise. Specifically, for each sample that is originally labeled as entailment in the training set, we create an augmented sample by exchanging the premise and the hypothesis. Note that we avoid augmenting the case where *P* and *H* mutually *entail* each other because the new premise still entails the hypothesis after the exchange. To achieve this, we exclude an exchanged sample from relation augmentation if it is still identified as entailment by a pretrained model netuned on SNLI (Bowman et al., 2015). In terms of the augmented samples, the program receives a positive reward during training if and only if it reaches the

correct final state *reverse entailment* $p \bullet q$.

4.2.3 Introspective Revision

The key challenges of developing interpretable neural natural logic models include coping with spurious reasoning paths (incorrect paths $r = [r_t]_{t=1}^{m}$ leading to the correct inferential label for a premise-hypothesis pair) as well as training inefficiency. Finding a correct program that reaches the correct label is challenging because it is inefficient to explore a space of 5^m paths for a reward. A positive reward for the correct path is often sparse.

We propose to use the fail-and-fix approach based on the newly proposed Back-Search algorithm (Li et al., 2020) to mitigate training inefficiency caused by sparse positive rewards, which, based on a failed program that earns no positive reward, searches for better proof paths in its neighborhood that reaches the correct final prediction. To solve the spurious issue in this fail-and-fix framework, we propose Introspective Revision, which leverages external commonsense knowledge (denoted as K) to control spurious proof paths. We believe unstated commonsense knowledge is important not only for improving prediction accuracy (which, as discussed in Sec. 2, often results from fitting to spurious correlations), but critical for developing interpretable natural language reasoning models by avoiding spurious proofs.

Without loss of generality, we distinguish a non-spurious program r from spurious ones based on the following assumption, whose effectiveness will be shown and discussed in our experiments.

Assumption 4.2.1. *A program r has a larger probability than another program r' to be a non-spurious program if r has a better agreement with the external knowledge base K.*

External Knowledge: Previous work (Chen et al., 2017a) queries the knowledge base for each pair of words between a premise and hypothesis exhaustively, which is inefficient

and likely to introduce undesired local relations. As a remedy, we found that the lightweight text alignment tool JacanaAligner (Yao et al., 2013), though not accurate enough to align all pairs of associated phrases in the input, can be used to guide the search. For a hypothesis phrase s, we first apply JacanaAligner to obtain its associated premise phrase $\boldsymbol{s}$, and then query the WordNet (Miller, 1998) database for the possible natural logic relations for the phrase pair $\langle s; \boldsymbol{s}\rangle$:

① **Equivalence** ( ): $s = \boldsymbol{s}$ or $s \in \boldsymbol{s}$;

② **Forward Entailment** ($\in$): $s \in \boldsymbol{s}$ or $s = \boldsymbol{s}$ or $\exists u \in s; v \in \boldsymbol{s}$ and u is a hypernym of v;

③ **Reverse Entailment** ($\ni$): $\boldsymbol{s} \in s$ or $\exists u \in s; v \in \boldsymbol{s}$ and v is a hypernym of u;

④ **Alternation** (|): $\exists u \in s; v \in \boldsymbol{s}$ and u is an antonym of v;

where $u; v$ denote tokens in the phrase and $s \in \boldsymbol{s}$ means that s is a sub-phrase of $\boldsymbol{s}$. The local relations suggested by the knowledge base are formulated as a set of triplet proposals $\{t; r_t; p_t(r_t)\}$, where t is the time step, r_t is the suggested relation, and $p_t(r_t)$ is the model predicted probability that corresponds to r_t.

Human-curated rules, which are designed to retrieve natural logic relations from the knowledge base, are often imperfect. They inevitably introduce errors due to language variations. For example, intuitively $s \in \boldsymbol{s}$ indicates *forward entailment* ($\in$); e.g. "*white cat*" entails "*cat*", while there are cases where the sub-phrase rule indicates *equivalence* ( ); e.g.,"*have a chat with*" is equivalent to "*chat with*" in meaning. In rare cases, the relation can be *alternation* (|); e.g. "*fake gun*" and "*gun*" are distinct concepts. While $s \quad \boldsymbol{s}$ often indicates *equivalence* ( ), our rules need to handle cases where the adverbial is posed in separate phrases; e.g. "*a bike*" and "*near the park*" entails "*a bike*".

To deal with this issue, instead of making an intensive effort to design sophisticated rules to pinpoint a single accurate relation, we design relatively coarse rules to narrow

down the possibilities and leave the nal choice to the model. Speci cally, at each step, we provide the model with multiple possible candidates, and the proposed introspective revision algorithm introduced in this section decides to accept a useful proposal or reject a misleading one, based on both the reasoning objective (i.e. the label) and the predicted relation distribution.

Algorithm: Given a program r tr_tu$_t^m$ $_1$, the goal of the Introspective Revision algorithm is to nd a program r in the neighbourhood of r that executes to the correct answer y while maintaining a large agreement with the external knowledge K, as detailed in Algorithm 1. The algorithm starts with knowledge-driven revision (line 2 15). We arrange the triplet proposals obtained from the knowledge base as a priority queue tp$t; r_t; p_t$r$r_t$sq | 0 t ¤ $m; r$ P Bu. In each iteration the queue pops the triplet with the largest probability p_trr_ts that speci es a modi cation to the sampled program r^1 Fixp$r; t; r_t$q. In other words, changing the relation r_t at step t of program r to the proposed relation r_t yields a new program r^1. Following Li et al. (2020), the modi cation is accepted with a probability 1 if r^1 executes to the correct answer y; otherwise, it is accepted with a probability $\min$p1; p_trr_ts{p_tr$r_t$sq. The hyperparameter encourages the model to explore low-probability proposals. For each sample, the model accepts or rejects up to M triplets.

The knowledge-driven revision above is conservative because only the top-M proposals are considered. However, there are complex cases where the program still cannot reach the correct answer after M steps, or where the provided proposals are insu cient to solve the problem. In these cases, we apply the answer-driven revision (line 17 22) by conducting a 5 m grid search to nd modi cations that lead to the correct answers. Among the search results , we accept the triplet with the maximum probability. A detailed description of the grid search is presented in Algorithm 2.

Algorithm 1: Introspetive Revision

Input: program r $\{r_1, \ldots, r_m\}$, label y, relation proposals Φ
Param: maximum step M, threshold
Output: r

Init $r \leftarrow r$; $i \leftarrow 0$
/* Knowledge-driven revision. */
while $i < M$ *and* $\Phi \neq \emptyset$ **do**
  $(t, \tilde{r}_t, p_t(\tilde{r}_t)) \leftarrow \Phi$.pop()
  sample $u \sim U(0, 1)$
  $r^1 \leftarrow Fix(r, \tilde{r}_t, t)$
  /* Accept or reject $(t, \tilde{r}_t, p_t(\tilde{r}_t))$ */
  if $Execute(r^1) = y$ *and* $u <$ **then**
    $r \leftarrow r^1$
  else
    sample $u \sim U(0, 1)$
    if $u < min(1, \frac{p_t(\tilde{r}_t)}{p_t(r_t)})$ **then**
      $r \leftarrow r^1$
    end
  end
  $i \leftarrow i + 1$
end
/* Answer-driven revision. */
if $Execute(r) = y$ **then**
  $\Psi \leftarrow GridSearch(r, \Phi, y)$
  if $\Psi \neq \emptyset$ **then**
    $(t, \tilde{r}_t, p_t(\tilde{r}_t)) \leftarrow \Psi$.pop()
    $r \leftarrow Fix(r, \tilde{r}_t, t)$
  end
end
Return: r

Following the reward in Eq. 4.7 and the objective function in Eq. 4.8, we compute a new objective function J^1 with the modified program r and its corresponding reward R.

Algorithm 2: GridSearch

Input: program r $\{r_1, \ldots, r_m\}$, label y, relation proposals Φ
Output: Ψ

```
Init Ψ ← PriorityQueue()
for t ← 1 to m do
    foreach r̃ ∈ Φ do
        r^t ← Fix(r, r̃, t)
        if Execute(r^t) = y then
            Ψ.push((t, r̃, p_t(r̃)))
        end
    end
end
if Ψ ∩ Φ = ∅ then
    Ψ ← Ψ ∩ Φ
end
Return: Ψ
```

The model learns by optimizing the hybrid training objective J_{hybrid}, defined as:

$$J^1 = \sum_{t=1}^{m} \log p_t(r_t) \cdot R_t \tag{4.9}$$

$$J_{hybrid} = \alpha J + (1 - \alpha) J^1, \tag{4.10}$$

where α is a weight that specifies the importance of the revision. The introspective revision algorithm is only applied during training since the label y is required to determine whether a proposal is accepted or not.

4.3 Experiments Setup

4.3.1 Statistics for Introspective Revision

In Table 4.1, we present the statistics for the introspective revision at the start/end of the training, where the natural logic programs are sampled from the predicted distribution. Approximately 80% of the samples perform at least one step of revision, and at the end

Phase	Revision (Knowl. / Answ. / Both)	Success Rate of Revision
Start	80.4% (85.2% / 8.1% / 6.7%)	59.4%
End	81.7% (80.3% / 5.9% / 13.8%)	98.4%

Table 4.1: The percentage of samples being revised and the revision success rate.

Relation	Knowledge Available	Knowl.-driven start / end	Answ.-driven start / end
Equivalence	1.035	0.595 / 0.482	0.096 / 0.026
Fwd. Entail	1.087	0.370 / 0.523	0.014 / 0.037
Rev. Entail	0.249	0.097 / 0.191	0.008 / 0.061
Alternation	0.012	0.004 / 0.008	0.001 / 0.037
Sum	2.383	1.066 / 1.204	0.119 / 0.161

Table 4.2: The average number of triplet proposals obtained and accepted.

of the training, there is an increasing chance (98.4% vs. 59.4%) that introspective revision helps the model reach the nal correct NLI prediction. In Table 4.2, we show the statistics of the average number of triplet proposals obtained from WordNet and the average number of proposals accepted by knowledge or answer-driven revision during training. *Equivalence* ( ) and *forward entailment* (€) make up a large portion of the proposals, while the *alternation* relation is scarce due to the sparsity of the antonym relation obtained from WordNet. As a result, the numbers of proposals accepted in the knowledge-driven revision are imbalanced across di erent relations. Moreover, we found that the number of accepted answer-driven revisions slightly increased at the end of the training, which is because as the training proceeds, the programs produced by the model are closer to the target labels.

4.3.2 Datasets for Monotonicity Reasoning

Though SNLI provides a large high-quality training resource for natural language inference, it contains undesired bias (Gururangan et al., 2018) and the samples rarely come with

downwards monotonicity inferences. In this chapter, we evaluate the proposed model in multiple datasets that involves monotonicity reasoning.

The HELP dataset has 35,891 inference pairs, and the lexical substitutions are automatically generated according to WordNet (Miller, 1998). To avoid unnatural sentences generated by the automatic process, MED provides 5,382 inference pairs, which are either written by crowd-workers or manually collected from linguistics publications. MoNLI contains 2,678 naturalistic sentence pairs focused on the compositional interactions between lexical entailment and negation.

In the above datasets, a premise and the corresponding hypothesis di er by *1-hop*; i.e., they are di erent by either a lexical substitution, insertion, or deletion. In addition, we also evaluated our model on the Natural Logic 2-hop dataset (Feng et al., 2020), which requires a model to perform a 2-hop natural logic composition according to Table 2.3.

4.3.3 Dataset for Systematicity of Monotonicity Reasoning

Making systematic generalizations from limited data is an essential property of human language (Lake and Baroni, 2018). While ne-tuning pretrained transformers achieve high NLI accuracy, Yanaka et al. (2020) have recently shown that these models have limited capability of capturing the systematicity of monotonicity inference. We use the dataset proposed by Yanaka et al. (2020) to evaluate the model's ability in compositional generalization: the model is exposed to all primitive types of quanti ers Q and predicate replacements R, but samples in the training set and test set contain di erent combinations of quanti ers and predicate replacements. Speci cally, with an arbitrarily selected set of quanti ers t*q*u and predicate replacement t*r*u, the training set contains data $D^{tqu;\boldsymbol{R}} \cup D^{\boldsymbol{Q};tru}$ while the test data only includes the complementary set $D^{\boldsymbol{Q}\setminus tqu;\boldsymbol{R}\setminus tru}$. An example of compositional generalization is shown below:

(1) *P: Some dogs run* ñ *H: Some animals run*

(2) *P: No animals run* ñ *H: No dogs runs*

(3) *P: Some small dogs run* ñ *H: Some dogs run*

An ideal model can learn from the training samples (1), (2), and (3) the entailment relations between concepts *small dog* € *dog* € *animal*, as well as the fact that the quanti er *some* indicates the upward monotonicity and *no* indicates the downward. During testing, the model needs to compose the entailment relations and the monotonicity signatures to make inference over unseen combinations, e.g., sample (4):

(4) *P: No dogs run* ñ *H: No small dogs run*

(5) *P: Near the shore, no dogs run* ñ
H: Near the shore, no small dogs run

To test the model stability, Yanaka et al. (2020) also added adverbs or prepositional phrases as test-only noise to the beginning of both the premise and the hypothesis, e.g., sample (5). To evaluate the model's ability for systematic compositional generalization, all models are trained on the designated training split and tested on the complementary test split, exactly following the data splits in Yanaka et al. (2020).

4.3.4 Benchmark for Rationales

To evaluate the model interpretability, we derive the predicted rationales from the natural logic programs and compare it with human annotations in e-SNLI (Camburu et al., 2018). Speci cally, our model regards as rationales the hypothesis phrases s_t that satis es: (1) z_t points to the nal prediction according to the grouping described at the end of Sec. 4.2.2; (2) z_t $z_{t\ 1}$. Following DeYoung et al. (2020), we use Intersection Over Union (IOU) formulated in Eq. 4.11 as the evaluation metric: the numerator is the number of shared tokens between the model-generated rationales and the gold rationales, and the denominator

is the number of tokens in the union. We also compute ner-grained statistics over individual rationale phrases. Following DeYoung et al. (2020), a predicted rationale phrase p matches an annotated rationale phrase q when $IOU(p,q) \geq 0.5$, and we use *precision*, *recall* and *F1 score* to measure the phrasal agreement between the predicted rationales and human annotations. We also invited 3 graduate students (not the authors of this thesis) to evaluate the quality of the predicted rationales on the rst 100 test samples in e-SNLI. Given the premise-hypothesis pair and the golden label, the evaluators judged the explanation as plausible if the predicted rationale (1) alone is su cient to justify the label, and; (2) does not include the whole hypothesis sentence.

$$IOU \quad \frac{\textit{num-tokens}(R_{pred} \cap R_{truth})}{\textit{num-tokens}(R_{pred} \cup R_{truth})} \tag{4.11}$$

From the perspective of natural logic, we follow (Feng et al., 2020) to evaluate the quality of the natural logic programs. For each sample, the Natural Logic 2-hop dataset provides the gold program execution states, and we evaluated the accuracy of our predicted states $\hat{z}_t$ against the ground truth.

4.3.5 Implementation Details

Our model is trained on Stanford Natural Language Inference (SNLI) (Bowman et al., 2015), in which the relation between a premise and hypothesis is classi ed to either *entailment*, *contradiction*, or *neutral*. We set the unit reward 1.0, and optimize our model with Adam gradient descent for six epochs with a learning rate of 2e-5. We compare the models with discount factor $\in \{0.25, 0.50, 0.75, 1.00\}$ and $\in \{0.05, 0.10, 0.20\}$. We found that the test accuracies are not sensitive to when ≥ 0.50, and we select 0.50, 0.20, which achieved the best validation accuracy on SNLI. For the introspective revision algorithm we set M 3 based on the average number of proposals (2.383 proposals/sample) in Table 4.2.

Model	SNLI	HELP	MED	MoNLI	NL-2hop
ESIM (Chen et al., 2017b)	88.0	55.3	51.8	63.9	45.1
BERT-base (Devlin et al., 2019b)	90.1	51.4	45.9	53.0	49.3
GPT-2 (Radford et al., 2019)	89.5	52.1	44.8	57.5	48.3
Feng et al. (2020)	81.2	58.2	52.4	76.8	60.1
Ours – full model (A0)	87.5	**65.9**	**66.7**	**87.8**	**62.2**
w/o knowledge ①	87.2	62.8	62.2	77.0	61.7
w/o knowledge ②	87.4	65.8	64.2	81.7	51.7
w/o knowledge ③	87.5	65.6	65.9	83.6	61.6
w/o knowledge ④	87.6	65.4	64.7	83.3	58.2
w/o knowledge ①②③④ (A1)	87.6	65.0	64.8	77.3	48.8
w/o answer driven revision (A2)	87.5	65.4	65.5	85.1	60.9
w/o introspective revision (A3)	87.6	62.1	60.7	74.4	53.3
w/o relation augmentation (A4)	87.8	59.6	54.7	74.7	59.9
Ours w/ masked attention (A5)	75.9	61.3	61.6	70.9	54.6

Table 4.3: Model accuracy on multiple challenging test datasets.

We treat the revised program and the original program as equally informative by setting 0:5. Our code is available at `https://github.com/feng-yufei/NS-NLI`.

4.4 Experiments

We evaluate the performance of the proposed model on six NLI tasks from various perspectives: the ability to perform monotonicity inference (Sec. 4.4.1), reasoning systematicity (Sec. 4.4.2), and model interpretability (Sec. 4.4.3).

4.4.1 Performance on Monotonicity Reasoning

We conduct experiments on multiple recently proposed challenging test datasets for monotonicity inference: HELP (Yanaka et al., 2019b), MED (Yanaka et al., 2019a), and Monotonicity NLI (MoNLI) (Geiger et al., 2020). Unlike SNLI, half of the samples in HELP, MED, and MoNLI are in downward monotone, and they are categorized as *entailment* or *non-entailment*.

We compare our model with popular natural language inference baselines including ESIM (Chen et al., 2017b), BERT-base (Devlin et al., 2019b), GPT-2(Radford et al., 2019), and (Feng et al., 2020). Following Yanaka et al. (2019a) and to ensure a fair comparison, all models are trained on SNLI, and during testing, we regard *contradiction* and *neutral* as *non-entailment* if a binary prediction is required.

Table 4.3 shows the test accuracy on SNLI and four challenging test datasets. Our model performs consistently and signi cantly better than previous state-of-the-art models on all challenging datasets while achieving competitive "*in-domain*" performance on SNLI. Manual inspection shows that compared to GPT-2, a signi cant proportion of the failure cases on SNLI are due to errors from the projectivity parser, and the ambiguity between *contradiction* and *neutral* (Bowman et al., 2015). The introspective revision algorithm signi cantly boosts the model performance on the monotonicity reasoning test sets (A0 vs. A3). Ablation shows that the knowledge-driven revision improves the performance on MoNLI and the 2-hop dataset (A0 vs. A1), which suggests that without proper constraints, the answer-driven revision can lead to spurious reasoning. We found that removing *equivalence* ( ) (knowledge ①) from the knowledge-driven revision lowers the performance, because in this case the knowledge-driven revision mistakenly encourages the model to replace *equivalence* ( ) with *forward entailment* (€), which may lead to incorrect prediction under downward monotonicity. Compared to *forward entailment* (€) (knowledge ②), removing *reverse entailment* (•) (knowledge ③) and *alternation* (|) (knowledge ④) does not significantly a ect the results. We deduce that the relative importance of di erent relations are a ected by the frequency of the external knowledge, and without the help of the knowledge-driven revision, the model can still learn the *reverse entailment* (•) relation from relation augmentation in Sec. 4.2.2. The performance drops when the relation augmentation is vacant (A0 vs. A4).

We also include the model that masks both the past and the future hypothesis chunks in the transformer attention layers for local relation prediction (A5). The model with

masked attention yields signi cantly lower performance on SNLI, partly because aggressively masking the past hypothesis chunks changes the structure of the pretrained GPT-2 model, and thus the model bene ts less from the pretrained representations.

4.4.2 Systematicity of Monotonicity Inference

In Table 4.4, all models are trained with 3,270 samples and tested on the complementary test set with about 9,112 examples, exactly following the data split in Yanaka et al. (2020). While all baseline models achieved high training accuracy, BERT has limited performance on the test set. For our model, there is only a 3% gap between the training and test performance, which demonstrates that our model successfully learns to identify and compose the natural logic relations of the predicate replacements with limited training examples.

We also compare our model to variants of BERT and GPT-2 models that are aware of the token projectivity (models with Ö in Table 4.4). Speci cally, for each token, we concatenate the hidden states in the nal layer of the transformer with its projectivity feature. We aggregated the concatenated features with multiple feed-forward layers and applied average pooling before sending them to the classi cation layer. Results show that BERT and GPT-2 do not bene t from the projectivity features. The test accuracy drops with additional adverbs and preposition phrases, leaving space for future research on the robustness of unseen perturbations.

4.4.3 Evaluation of Model Explainability

The proposed model provides built-in interpretability following natural logic | the execution of programs $t z_t u_t^m{}_1$ (Eq. 4.6) provides explanation along with the model's decision making process, namely giving a *faithful* explanation (Jacovi and Goldberg, 2020).

We compare our model with representative neural rationalization models proposed by Lei et al. (2016), which learns to extract rationales without direct supervision, and Feng

Model	Train	Test	Test_{adv}	Test_{pp}
BERT-base	100.0	69.2	50.8	49.3
GPT-2	100.0	25.6	35.6	35.4
BERT-base$^{\ddot{O}}$	100.0	65.4	51.4	52.7
GPT-2$^{\ddot{O}}$	100.0	28.1	35.1	39.6
Ours w/o. IR	91.3	79.3	57.1	54.0
Ours	98.4	95.1	61.0	61.5

Table 4.4: Results for compositional generalization.

Model	e-SNLI IOU	e-SNLI Precision / Recall / F1	e-SNLI Human Eval.
Lei et al. (2016)	0.42	0.37 / 0.46 / 0.41	56 / 100
Feng et al. (2020)	0.27	0.21 / 0.35 / 0.26	52 / 100
Ours – full model (B0)	**0.44**	**0.58** / **0.49** / **0.53**	**80** / 100
w/o. external knowledge (B1)	0.41	0.53 / 0.45 / 0.48	67 / 100
w/o. introspective revision (B2)	0.40	0.52 / 0.43 / 0.47	68 / 100
w/o. relation augmentation (B3)	**0.44**	0.57 / 0.48 / 0.52	75 / 100
Ours – BERT encoder (B4)	0.14	0.20 / 0.15 / 0.17	29 / 100

Table 4.5: Evaluation for the model generated explanation.

Model	Natural Logic 2-hop Acc.
Lei et al. (2016)	–
Feng et al. (2020)	0.44
Ours – full model (B0)	**0.52**
w/o. external knowledge (B1)	0.44
w/o. introspective revision (B2)	0.43
w/o. relation augmentation (B3)	0.51
Ours – BERT encoder (B4)	0.28

Table 4.6: Evaluation for the model generated explanation.

et al. (2020), which explains its prediction by generating natural logic reasoning paths. The summary statistics in Table 4.5 shows that our model matches Lei et al. (2016) on the IOU score, and that it produces rationales with signi cantly higher precision and F1 scores on the e-SNLI test set. The superior rationalization performance is also supported by the human evaluation mentioned above (the 4th column in Table 4.5). Compared to Feng et al. (2020), our model produces intermediate natural logic states that better agree with the ground truth. The results in Table 4.5 show that the model explanation signi cantly bene ts from the external knowledge (B0 vs. B1), and the answer-driven revision alone does not improve the quality of the generated rationales (B1 vs. B2). We also compare our model to the system that replaces the uni-directional attention model GPT-2 with the bi-directional attention model BERT. The model with a BERT encoder yields signi cantly lower scores on interpretability (B0 vs. B4).

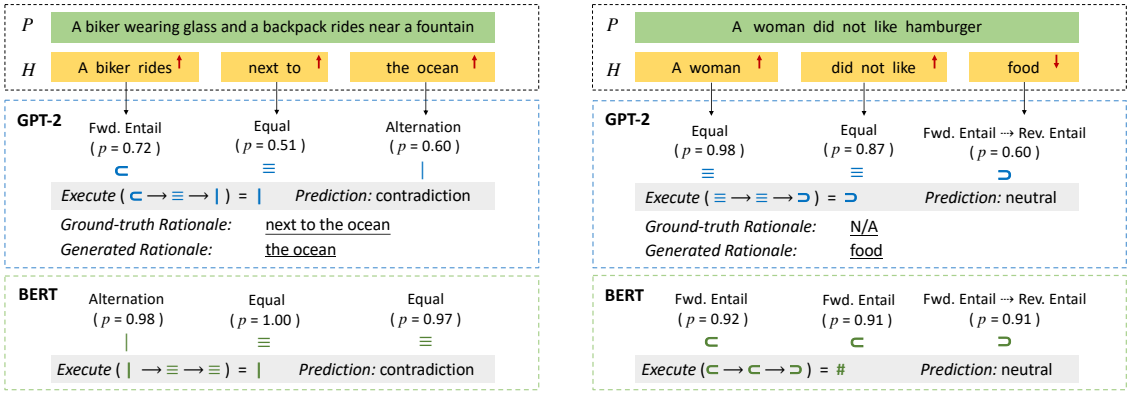

Figure 4.2: Examples for predictions and explanation for some cases from SNLI (left) and MoNLI (right).

4.4.4 Case Study

The upper part of the Fig. 4.2 shows how our natural logic model makes predictions during testing. The left example involves upward monotone. Upon seeing the premise and the rst hypothesis phrase *A biker rides*, the model predicts the local relation as *forward entailment*

(r_1 `€') at time step $t$ 1. The predicted relation stays unchanged after applying the projection function p`€'q `€' because it is in the context of upward monotone. According to Table 2.3 we have $z_1$ $z_0$ b $r_1$ `€'. Similarly, as the second prediction for the phrase *next to*, relation *equivalence* (r_2 ` ') does not change the reasoning states because $z_2$ $z_1$ b $r_2$ `€'. The third hypothesis phrase *the ocean* is a distinct concept against *a fountain* in the premise, our model outputs relation *alternation* (r_3 `|') and we have $z_3$ $z_2$ b $r_3$ `|'. The model runs out of the hypothesis phrases after 3 steps, and reaches *contradiction* according to the nal state z_3.

An additional example with downward monotone is illustrated on the right of Fig. 4.2. Our model predicts the relation *forward entailment* (r_3 `€') at the third time step since *food* includes *hamburger*. The projection function ips the relation to *reverse entailment* (•) because according to the projectivity in Table 2.2, the rst argument that follows negation *did not* is in downward monotone, i.e., p`€'q `•'.

At the bottom of Fig. 4.2, we provide examples of the reasoning processes produced by the natural logic model that is built upon the bi-directional attention model BERT. Although it produces the same nal labels as our proposed model, the model based on BERT can predict wrong local relations due to its entangling e ect. Speci cally, the model with bi-directional attention is prone to make its nal decision in the rst place (82% of the cases in the human evaluation), and then predict local relations that can keep the initial decision during the program execution (according to the composition rules in Table 2.3). In the rst example in Fig. 4.2, to keep the rst predicted relation *alternation* (|) unchanged during execution, the model subsequently predicts a series of *equivalence* ( ) relations. In the second example, the model predicts local relation *forward entailment* (€) for each hypothesis phrase, and at the last step, the *forward entailment* (€) relation is projected to *reverse entailment* (•) according to the projectivity.

4.5 Summary and Future Works

4.5.1 Summary of the Contributions

In Chapter 4, we discussed the limitations of the end-to-end natural logic model proposed in Chapter 3. Then we proposed a neuro-symbolic approach that yields better NLI predictions and natural-logic-based explanations.

We designed a neuro-symbolic model that incorporates a pretrained transformer network and the natural logic program, which is a parameter-free state-transition program designed to compose neural network predictions to obtain the nal answer. Based on the transformer network and the natural logic program, we designed rewards and proposed to use the policy gradient method for model training.

We also designed an introspective revision algorithm, which greatly improves the symbolic reasoning ability of our proposed model. The introspective algorithm is also a novel method that plays an important role in incorporating commonsense knowledge bases into neuro-symbolic training.

We also formally stated the `*entangling*' problem, which was discovered by us during model development in Chapter 3 and Chapter 4. We gave a detailed explanation of why this problem happens and why the entangling problem will harm the model's interpretability. We also provide two solutions to alleviate the entangling problem.

In addition, we designed experiments proving the model's ability for monotonicity reasoning and provided natural logic explanations. We also showed that our neuro-symbolic model is a practical solution to the compositional generalization problem.

4.5.2 Potential Extensions

The neuro-symbolic natural logic model achieved competitive performance on NLI, and it provided model explanations based on natural logic. Though natural logic is designed to

solve the natural language inference problem, the idea behind it can be generalized to other applications. In the next chapter, we will discuss how to extend part of the natural logic proof (i.e., step-wise reasoning) to the multi-hop question answering problem, where the model is asked to verify or find an answer to a statement or query, given a collection of supportive text documents.

Chapter 5

Extending Neuro-Symbolic Methods to Multi-Hop Question Answering

5.1 Introduction

Although natural logic is a proof logic designed especially for natural language inference (NLI), some of its central motivations deserve deeper attention from researchers of related fields. One of the central ideas behind natural logic is span-level step-wise reasoning. Remember in the previous chapters, we discussed how the neuro-symbolic natural logic model predicts local relations and composes them step by step to reach the final answer. The step-wise reasoning can be applied to many other natural language processing tasks. For example, the multi-hop question answering (QA) task requires models to answer a given question by gathering information from multiple pieces of text evidence. In many cases, the process of reasoning through a chain of evidence follows a step-wise manner, where a human identifies each span of interest one by one and yields a final answer after all evidence is checked. As a result, it would be interesting to investigate what kind of benefit a step-wise reasoning model can bring over current deep learning research.

In addition to the step-wise reasoning paradigm, natural logic literature proposes the idea of surface-form reasoning (Section 2.4), where, instead of parsing a query to a logic-form

program, a model performs reasoning over the pure text form. Nowadays, most parsing-and-execution models (Mao et al., 2019; Khot et al., 2020) are out-performed by the end-to-end neural models because the parsing process introduces errors, and often the domain-speci c language behind the parser has limited power to scale.

In this chapter, we explore how to extend the neuro-symbolic method proposed in previous chapters, and especially the idea of step-wise reasoning, to multi-hop QA applications (Yang et al., 2018; Welbl et al., 2018; Thorne et al., 2018; Chen et al., 2019a; Fang et al., 2020; Mishra et al., 2021). The QA problem we study in this chapter aims to answer a given text question Q by referring to a supportive document D t$P_1; P_2; :::; P_n$u, where P_i are di erent pieces of text (i.e., sentences or paragraphs) in the document. In this chapter, we use *Ans* to represent the answer, which can be either a word, a phrase, or a sentence depending on the question Q. For the multi-hop QA task, to answer the question Q, a model needs to acquire at least two pieces of information (passages) from the document D. A multi-hop example from the HotpotQA dataset (Yang et al., 2018) is shown as follows:

- Q: *Three Men on a Horse* is a play by a playwright born in which year?
- Ans: 1887.
- P_1: *Three Men on a Horse* is a play by George Abbott and John Holm ...
- P_2: : George Francis Abbott (June 25, 1887 { January 31, 1995) was an American playwright ...
- P_3: *Three Men on a Horse* made its debut in ...
- ...

In the preceding example, Passage P_1 and P_2 are the required evidence for the question Q, and other passages like P_3 are not required when answering the question. A typical multi-hop QA model (Zhuang and Wang, 2019; Yichen Jiang and Bansal, 2019; Tu et al.,

2020; Fang et al., 2020) will first learn to retrieve the required evidence passages based on the question, then perform text reasoning over the retrieved passages to find the correct answer *Ans*.

The multi-hop QA problem can be formulated as an extension to NLI (Mishra et al., 2021): the document D serves as the premise, and the question-answer pair r$Q; Ans$s are treated as the hypothesis[1]. However, the multi-hop QA problem has unique features compared with NLI. On the one hand, in QA tasks, the premise (the document D) is usually much longer than the hypothesis because only a very small amount of the sentences in the document are relevant to the question and the answer. For example, in the aforementioned case, only P_1 and P_2 are relevant while other passages are not useful to the question. On the other hand, the QA task does not emphasize the difference between strict entailment and equivalence. The question and answer can form the same sentence in D, or they can form a paraphrased or entailed one.

Similar to the neuro-symbolic model introduced for NLI, we aim to build a model that simultaneously solves the QA problem and provides a faithful explanation for its decisions. We extend ideas of the neuro-symbolic natural logic model to QA tasks: for each piece of text in the r$Q; Ans$s statement, the model needs to find the relevant sentences in D and determine the natural logic relations between the piece and its relevant sentences. A r$Q; Ans$s pair is valid (i.e., supported by the documents) if all its text pieces are entailed by the document D, and the pair is not valid if at least one of the pieces is not supported by D.

In this chapter, we introduce two pivotal works on the path of our objective, one as the model for automatic selection of relevant information in the document D (Section 5.2), and the other as the complete pipeline of question answering (Section 5.3).

[1] In this chapter, we limit our focus to the QA problems that have already provided the candidate answers (correct or incorrect) in a pool, and we leave the generative QA tasks for the future works. In many QA samples, the r$Q; Ans$s pair can be converted to a sentence statement.

5.2 Learning to Discover Reasoning Chain in Multi-Hop Question Answering

In this section, we proposed a model for the automatic selection of relevant information (relevant passages or sentences) in the document D. Although there are recent interests in predicting reasoning chains for multi-hop QA (Asai et al., 2019; Chen et al., 2019a; Ding et al., 2019; Xiao et al., 2019; Tu et al., 2020), they all consider a fully supervised setting, that is, the annotated reasoning chains are available during training. The pretrained transformers achieved high accuracy on relevant passage selection in a supervised setting, but in real applications, it is very expensive to annotate the relevant passages or sentences, compared to annotating the question-answer pairs only (the question-answer pairs and the supporting documents can be automatically generated from existing knowledge graphs and the Wikipedia documents, but it is expensive to let human beings pinpoint the exact sentences that support the questions and answers). In this section, the task of discovering reasoning chains is essentially an unsupervised problem, given we have no access to the annotated reasoning chains.

Moreover, even though we can train a model that selects relevant passages with high accuracy, how can we further explore the semantic connections between the relevant passages? Taking the QA pair shown in Section 5.1 again as an example, *George Abbott* is the person who connects P_1 and P_2, which plays an important role in the reasoning process of the given question. The connection among the question, relevant passages, and the answer is crucial in explaining the QA decisions, and in this section, we are developing models that simultaneously search for relevant passages and the links between the passages. To illustrate the proposed model, we need to de ne the concept of reasoning chains.

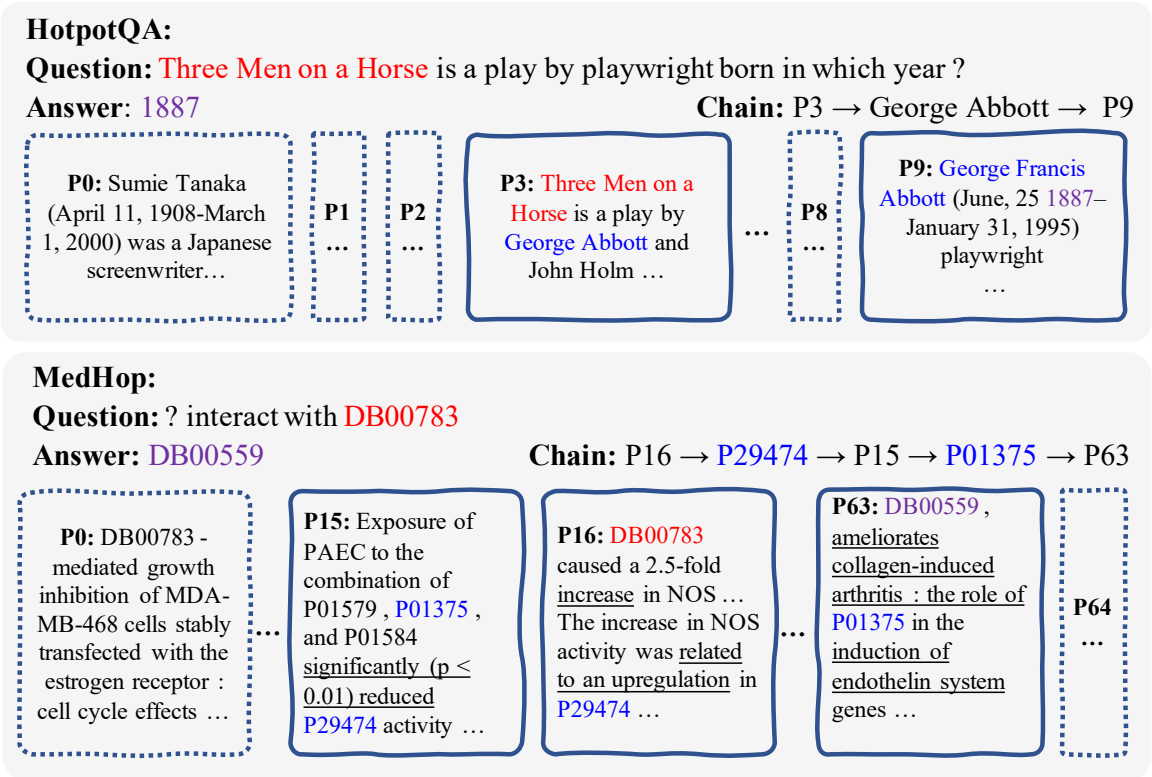

Figure 5.1: An example of reasoning chains in HotpotQA (2-hop) and MedHop (3-hop).

5.2.1 Task Definition

Reasoning Chains. Examples of reasoning chains in HotpotQA and MedHop are shown in Figure 5.1. Formally, we aim at discovering reasoning chains in the form of $\langle P_1 \rightarrow e_{1;2} \rightarrow P_2 \rightarrow e_{2;3} \rightarrow \cdots \rightarrow e_{n-1;n} \rightarrow P_n \rangle$, where each P_i is a passage and each $e_{i;i-1}$ is an entity that connects P_i and P_{i-1}, that is, appearing in both passages. The last passage P_n in the chain contains the correct answer. We say P_i connects $e_{i-1;i}$ and $e_{i;i-1}$ in the sense that it describes a relationship between the two entities. Taking the HotpotQA question and answer in Figure 5.1 as example, the desired reasoning chain is $\langle P_3 \rightarrow George\_Abbott \rightarrow P_9 \rangle$.

Our Task. Given a QA pair $\langle Q; Ans \rangle$ and a collection of passages D, we can extract all possible candidate chains that satisfy the aforementioned conditions, denoted as C. The

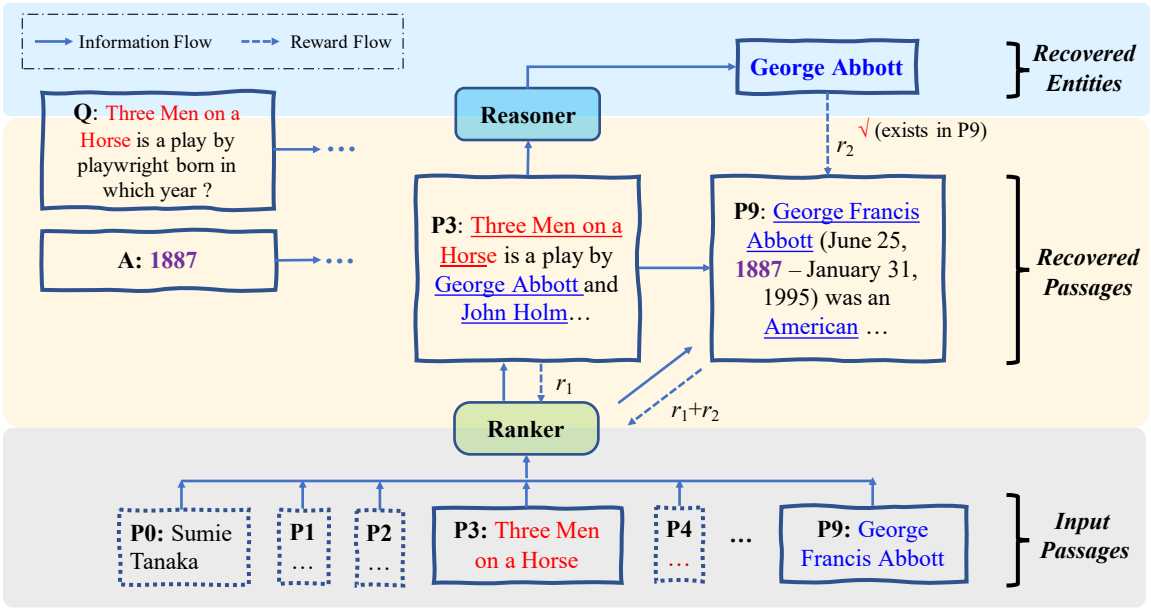

Figure 5.2: Overview of the cooperative Ranker and Reasoner.

goal of reasoning chain discovery is to extract the correct chains from all the candidates, given $Q; Ans$ and D as inputs.

We build the reasoning chain discovery model with the noisy distant supervision signals from question-answer pairs. Figure 5.2 illustrates our overall framework. We first propose a conditional selection model that optimizes the passage selection by considering their orders (Section 5.2.2). We then propose a cooperative Reasoner-Ranker game (Section 5.2.3) in which the Reasoner discovers the linking entities that point to the next passage that Ranker selects. This encourages the Ranker to select chains with patterns that are easy for a model (Reasoner) to capture. The ranker and reasoner collaborate to help each other de-noise the distant supervision signals and discover reasoning chains with entity information.

5.2.2 Passage Ranking Model

The key component of our framework is the Ranker model, which is provided with a question Q and K passages D $\{P_1; P_2; \ldots; P_K\}$ from a pool of candidates, and outputs a chain of

selected passages.

MatchLSTM and Passage Scoring Given the question sentence embeddings $Q = \{q_0, q_1, ..., q_N\}$, and $H_i = \{\tilde{h}_{i,0}, \tilde{h}_{i,1}, ..., \tilde{h}_{i,M_i}\}$ of each passage $P_i \in D$, we use the MatchLSTM model (Wang and Jiang, 2016) to match Q and H_i as follows:

$$\begin{aligned} e_{jk} &= q_j^T \tilde{h}_{i,k} \\ \alpha_j &= \sum_{k=0}^{M} \frac{\exp(e_{jk})}{\sum_{l=0}^{M} \exp(e_{jl})} \tilde{h}_{i,k} \\ m_{i,j} &= [q_j, \alpha_j, q_j \odot \alpha_j, q_j - \alpha_j] \\ m_i &= \text{MaxPool}(\text{GRU}[m_{i,0}, m_{i,1}, ..., m_{i,N}]). \end{aligned} \tag{5.1}$$

The final vector m_i represents the matching state between Q and P_i. All the m_is are then passed to a linear layer that outputs the ranking score of each passage. We apply softmax over the scores to get the probability of passage selection $P(P_i|Q)$.

Intuitively, the MatchLSTM model (Wang and Jiang, 2016) computes the matching score between Q and each H_i. We denote the derived distribution of passage selection as $P(p_i|q) = \text{MatchLSTM}(H_i, Q)$ for simplicity.

Conditional Selection. To model passage dependency along the chain of reasoning, we use a hard selection model that builds a chain incrementally. Provided with the K passages, at each step t the Ranker computes $P^t(P_i|Q^{t-1}); i = 0, ..., K$, which is the probability of selecting passage P_i conditioned on the query and previous states representation Q^{t-1}. Then we sample one passage P^t according to the predicted selection probability.

$$\begin{aligned} p^t &= \text{Sampling}(P^t) \\ Q^t &= \text{FFN}([Q^{t-1}, m_P^t]) \\ P^{t-1}(P_i|Q^t) &= \text{MatchLSTM}(P_i, Q^t), \end{aligned} \tag{5.2}$$

The first step starts with the original question Q^0. A feed-forward network is used to project the concatenation of query encoding and selected passage encoding m_p^t back to the query space, and the new query Q^{t-1} is used to select the next passage.

Reward via Distant Supervision. We use policy gradient (Williams, 1992) to optimize our model. Because we have no access to annotated reasoning chains during training, the reward comes from distant supervision. Specifically, we reward the Ranker if a selected passage appears as the corresponding part of a distant supervised chain in C. The model receives an immediate reward at each step of selection.

In this thesis, we only consider chains consisting of ¤ 3 passages (2-hop and 3-hop chains).[2] For the **2-hop cases**, our model predicts a chain of two passages from the candidate set C in the form of P_h Ñ e Ñ P_t. Each candidate chain satisfies that P_t contains the answer, while P_h and P_t contain a shared entity e. We call P_h the head passage and P_t the tail passage. Let $P_T\{P_H$ denote the set of all tail/head passages from C. Our model receives rewards $r_h; r_t$ according to its selections:

$$r_t \quad 1{:}0 \quad \text{ð ñ} \quad P_t \text{ P } P_T; \; r_h \quad 1{:}0 \quad \text{ð ñ} \quad P_h \text{ P } P_H \tag{5.3}$$

For the **3-hop cases**, we need to select an additional intermediate passage P_m between P_h and P_t. If we reward any P_m selection that appears in the middle of a chain in candidate chain set C, the number of feasible options can be very large. Therefore, we make our model first select the head passage P_h and the tail passage P_t independently and then select P_m conditioned on $\text{p}P_h; P_t\text{q}$. We further restrict that each path in C must have the head passage containing an entity from Q. Then the selected P_m is only rewarded if it appears in a chain

[2] It has been shown that ¤ 3-hops can cover most real-world cases, such as KB reasoning (Xiong et al., 2017; Das et al., 2018).

in C that starts with P_h and ends with P_t:

$$\begin{aligned} r_h &= 1.0 \cdot \mathbb{1}[P_h = P_H], \; r_t = 1.0 \cdot \mathbb{1}[P_t = P_T] \\ r_m &= 1.0 \cdot \mathbb{1}[\text{path}(P_h, P_m, P_t) \in C] \end{aligned} \tag{5.4}$$

5.2.3 Cooperative Reasoner

To alleviate the noise in the distant supervision signal C, in addition to the conditional selection, we further propose a cooperative Reasoner model, also implemented with the MatchLSTM architecture, to predict the linking entity from the selected passages. Intuitively, when the Ranker makes more accurate passage selections, the Reasoner will work with less noisy data and thus is easier to succeed. Specifically, the Reasoner learns to extract the linking entity from chains selected by a well-trained Ranker, and it benefits the Ranker training by providing extra rewards. Taking 2-hop as an example, we train the Ranker and Reasoner alternatively as a cooperative game.

MatchLSTM for Reasoner Given the question embedding $Q^r = [\tilde{q}^r_0, \tilde{q}^r_1, \ldots, \tilde{q}^r_N]$ and the input embedding $H^r = [\tilde{h}^r_0, \tilde{h}^r_1, \ldots, \tilde{h}^r_M]$ of passage P, the Reasoner predicts the probability of each entity in the passage being the linking entity of the next passage in the chain. We use a reader model similar to (Yang et al., 2018) as our Reasoner network.

We first describe an attention sub-module. Given input sequence embedding $A = [\tilde{a}_0, \tilde{a}_1, \ldots, \tilde{a}_N]$ and $B = [\tilde{b}_0, \tilde{b}_1, \ldots, \tilde{b}_M]$, we define $M = \text{Attention}(A, B)$:

$$\begin{aligned} e_{jk} &= \tilde{a}_j^T \tilde{b}_k \\ \tilde{b}'_k &= \sum_{j=0}^{N} \frac{\exp(e_{jk})}{\sum_{l=0}^{N} \exp(e_{lk})} \tilde{a}_j \\ m_k &= \text{FFN}([b_k, \tilde{b}'_k, b_k \odot \tilde{b}'_k, b_k - \tilde{b}'_k]) \\ M &= [m_0, m_1, \ldots, m_M], \end{aligned} \tag{5.5}$$

where FFN denotes a feed forward layer that projects the concatenated embedding back to the original space.

The Reasoner network consists of multiple attention layers, together with a bidirectional GRU encoder and skip connection.

$$\begin{aligned} M_1^r &= \text{Attention}(Q^r, H^r) \\ \overrightarrow{H}_1^r &= \text{Bi-GRU}(M_1^r) \\ M_2^r &= \text{Attention}(H_1^r, \overrightarrow{H}_1^r) \\ \overrightarrow{H}^r &= [h_0^r, h_1^r, ..., h_M^r] = \text{Bi-GRU}(M_1^r, M_2^r) \end{aligned} \tag{5.6}$$

For each token $e_k, k = 0, 1, ..., M$ represented by h_k^r at the corresponding location, we have:

$$P^r(e_k|P) = \begin{cases} g(h_k^r), & \text{if } e_k \text{ is a named entity} \\ 0, & \text{otherwise} \end{cases}. \tag{5.7}$$

where g is the classification layer, softmax is applied across all entities to get the probability. We denote the aforementioned computation as $P^r(e_k|P) = \text{MatchLSTM.Reader}(e_k, P)$ for simplicity.

5.2.4 Cooperative Training

The Ranker and the Reasoner are trained cooperatively. **Reasoner Step**: Given the first selected passage P_t[3] selected by the trained Ranker, the Reasoner predicts the probability

[3] In practice, the Ranker selects P_t first because we found that for HotpotQA and Medhop, the number of candidate tail passages is smaller than that of the head passages, which makes the selection of P_t more accurate. The same method holds for selecting P_h first. Section 5.5.1 shows that starting from the answer is empirically better.

of each entity e appearing in p_t. The Reasoner is trained with the cross-entropy loss:

$$y_e \begin{cases} 1; & \text{if } e \in P_h \\ 0; & \text{otherwise} \end{cases} : \quad P(e|P_t) \quad \text{MatchLSTM\_Reader}(e; P_t) \tag{5.8}$$

Ranker Step: Given the Reasoner's top-1 predicted linking entity e, the reward for Ranker at the 2nd step is defined as:

$$r_h \begin{cases} 1; & \text{if } P_h \in P_H \\ r; & \text{if } e \in P_h; P_h \in P_H \\ 0; & \text{otherwise} \end{cases} \tag{5.9}$$

Figure 5.2 shows the cooperation between the Ranker and the Reader. The Ranker selects a passage P at each step conditioned on the question Q and history selection, and receives a reward r_1 if P is a piece of evidence. Conditioned on Q, the Reasoner predicts which entity in P links to the next evidence passage. The Ranker receives an extra reward r_2 if its next selection is connected by the entity predicted by the Reasoner. Both Q and answer Ans are model inputs. While Q is fed to the Ranker/Reasoner as input, empirically the best way of using Ans is for constructing the evidence passage set for computing rewards r_1.

The extension to 3-hop cases is straightforward: the Ranker selects P_h, P_t and the middle passage P_m first, then the Reasoner reads both the selected P_h and P_t to output two entities $e_h; e_t$. The Ranker receives one extra reward if $e_h; e_t$ exist in the selected passage P_m.

5.3 A Neuro-symbolic Approach for Medical Text Reasoning

In this section, we propose a neuro-symbolic multi-hop QA model for medical text reasoning. The model is designed to solve the Medhop challenge (Welbl et al., 2018), which is detailed as follows.

5.3.1 Task Definition

Given a query medicine Q and a collection of supportive document D, the model needs to select a target medicine Ans from a pool of candidates C that interacts with medicine Q. As is shown in the lower part of Figure 5.1, in the Medhop dataset, each medicine or protein has a unique identi er number.

The medicine interactions are described in the documents $D = \{P_1; P_2; :::; P_n\}$, where n is the number of passages in the documents. Each passages is an abstract of a medical journal paper that describes the interactions between several types of protein or medicines. The correct answer Ans interacts with the query Q through the following mechanism:

- There exists a passage $P_i \in D$, which describes the interaction between medicine Q and protein S.
- There exists a passage $P_j \in D$, which describes the interaction between protein S and protein T.
- There exists a passage $P_k \in D$, which describes the interaction between protein T and medicine Ans.

According to the dataset description (Welbl et al., 2018), the pool of candidates C only contains one medicine that interacts with the query Q. In most cases, the interaction can be explained by a 3-hop reasoning chain, that is, the interaction mechanism shown above. We show an example of medicine interaction in the lower part of Figure 5.1.

5.3.2 Pre-processing Reasoning Chains

As the first step, given the query Q and a candidate $c \in C$, we extract *potential* 3-hop reasoning chains from the document D. In this section, we denote the tuple (chain of passages and objects) $(Q, S, P_i, S, T, P_j, T, c, P_k)$ as a reasoning chain. We regard a potential reasoning chain as a valid reasoning chain, if, in each passage, the interaction between the co-occurring protein and medicine is proven to be true according to the text, that is, P_i confirms that Q and S have an interaction, P_j confirms that S and T have an interaction, and P_k confirms that T and c have an interaction. Regardless of whether the medicines and proteins interact or not, a potential reasoning chain satisfies the following properties:

1. $Q, S \in P_i$
2. $S, T \in P_j$
3. $T, c \in P_k$

The query medicine Q and the candidate c interact if and only if there exists at least one valid reasoning chain $(Q, S, P_i, S, T, P_j, T, c, P_k)$. The medicine Q does not interact with the candidate c if there is no such valid reasoning chain in the documents (this includes the cases where there is no potential chain that links Q and c, or one of the passages does not prove the interaction between the co-occurring protein and medicine).

When implementing the model, we perform an exhaustive search to find all potential reasoning chains for each candidate $c \in C$.

5.3.3 Passage Encoding and Predicting Interactions

The proposed model performs reasoning over the potential reasoning chains. In this chapter, we first explain how the passages and reasoning chains are encoded by the neural network, then we specify how the interactions are predicted.

Passage Encoding Given a passage P:

$$P = [x_1, x_2, ..., x_n], \tag{5.10}$$

where, x_i is the i-th token in the passage and n is the passage length. We encode the passage with recurrent neural network and self-attention:

$$h_i = BiLSTM(P, i) \tag{5.11}$$

$$e_{ij} = f(h_i)^T f(h_j) \tag{5.12}$$

$$\hbar_i = \sum_{j=1}^{m} \frac{\exp(e_{ij})}{\sum_{k=1}^{m} \exp(e_{ik})} h_j, \tag{5.13}$$

where $\hbar_i$ is the encoded contextualized representation of token x_i in the passage, which will be used later as the input of the interaction prediction layer.

Entity Encoding Before calculating the interaction between two proteins/medicines in the passage, we need to combine the contextualized token embedding with a trainable context-independent entity embedding. In Medhop, each protein/medicine has an unique identifier, for example, in Figure 5.1, the query medicine has the identifier DB00037. In this model, we map each protein/medicine in the dataset into a randomly initialized vector embedding, which is optimized during training. If two identical protein/medicine identifiers appear in different passages, they share the same entity encoding.

For a pair of protein/medicine S, T at location t_1, t_2 in the passage, we concatenate the contextualized token embedding $\hbar_{t_1}, \hbar_{t_2}$ with the entity encoding h_S, h_T:

$$\hbar_S = [h_S, \hbar_{t_1}] \tag{5.14}$$

$$\hbar_T = [h_T, \hbar_{t_2}], \tag{5.15}$$

where r ; s indicates concatenation. Based on the concatenated protein/medicine embedding, the probability of interaction between S and T can be predicted as follows:

$$p_{S;T;P} \quad \text{p} f \text{pr} \hbar_S ; \hbar_t \text{sqq}; \tag{5.16}$$

where f is the prediction layer and is the *sigmoid* activation function, $p_{S;T;P}$ denotes the probability of S interacting with T given the content of the passage P.

5.3.4 The Min-Max Objective

The proposed model is optimized by a min-max objective: for a speci c reasoning chain $\text{p}Q; S; P_i; S; T; P_j; T; c; P_k\text{q}$, the model will predict the probabilities for three di erent interactions $p_{Q;S;P_i}; p_{S;T;P_j}; p_{T;c;P_k}$. The reasoning chain indicates that Q interacts with c if and only if $p_{Q;S;P_i}; p_{S;T;P_j}; p_{T;c;P_k}$ are simultaneously large, thus whether the reasoning chain is valid is determined by the minimum value over three predicted interactions.

$$p\text{p} \quad \text{q} \quad min\text{pp}p_{Q;S;P_i}; \quad p_{S;T;P_j}; \quad p_{T;c;P_k}\text{q} \tag{5.17}$$

For a given candidate c, there can be m potential reasoning chains in the document, $_i$ P ; i 1; 2; :::; m. c interacts with Q as long as there is one valid reasoning chain . Thus the probability of c interacts with Q can be formulated as a maximum:

$$p\text{pc}\text{q} \quad \max_{i\text{P}} \quad p\text{p} \ _i\text{q} \tag{5.18}$$

We use cross-entropy loss as the objective function:

$$L \quad \sum_c \ log \ p\text{pc}\text{q} \quad \mathbb{1}\text{pc} \quad Ans\text{q} \tag{5.19}$$

5.4 Experiment Setup

5.4.1 Discover Reasoning Chains

Datasets. We evaluate our path selection model on HotpotQA bridge-type questions and on the MedHop dataset. In HotpotQA, the entities are pre-processed Wiki anchor link objects (which can be obtained from the original Wiki page of the passage), and in MedHop they are medicine/protein database identi ers.

For **HotpotQA**, two supporting passages are provided along with each question. We ignore the support annotations during training and use them to create ground truth on the development set: following Wang et al. (2019), we determine the order of passages according to whether a passage contains the answer. We discard ambiguous instances.

For **MedHop**, there is no evidence annotated. Therefore we created a new evaluation dataset by manually annotating the correct paths for part of the development set: we rst extract all candidate paths in the form of passage triplets $\mathrm{p}P_h; P_m; P_t\mathrm{q}$, such that P_h contains the query drug and P_t contains the answer drug, and $P_h\{P_m$ and $P_m\{P_t$ are connected by shared proteins. We label a chain as positive if all the medicine{protein or protein{protein interactions are described in the corresponding passages. Note that the positive paths are not unique for a question.

During training we select chains based on the full passage set D; at inference time we extract the chains from the candidate set C (see Section 5.2.1).

Baselines and Evaluation Metric. We compare our model with (1) random baseline, which randomly selects a candidate chain from the distant supervision chain set C; and (2) distant supervised MatchLSTM, which uses the same base model as ours but scores and selects the passages independently. We use accuracy as our evaluation metric. Because HotpotQA does not provide ground-truth linking entities, we only evaluate whether the

supporting passages are fully recovered (yet our model still outputs the full chains). For MedHop we evaluate whether the whole predicted chain is correct. We use Pennington et al. (2014a) as word embedding for HotpotQA, and Zhang et al. (2019) for MedHop.

In HotpotQA, on average we can nd 6 candidate chains (2-hop) in an instance, and the human-labeled true reasoning chain is unique. A predicted chain is correct if the chain only contains all supporting passages (exact match of passages).

In MedHop, on average we can nd 30 candidate chains (3-hop). For each candidate chain, our human annotators label whether it is correct or not, and the correct reasoning chain is not unique. A predicted chain is correct if it is one of the chains that humans labeled as correct.

The accuracy is de ned as the ratio:

$$acc \quad \frac{\#\text{ of correct chains predicted}}{\#\text{ of evaluation samples}} \tag{5.20}$$

5.4.2 Medical Text Reasoning

The medical text reasoning model is trained on the 1620 Medhop training cases for 100 epochs and evaluated on the 342 validation cases (the test split is not available because it is kept hidden by the leader-board provider). Because the training datasets are relatively small, we apply dropout layers to the recurrent neural network hidden states to prevent over- tting.

During experiments, we nd that the Medhop dataset is biased towards the name of the speci c medicines. This is because many medicines (chemical compounds) are highly active and may interact with a large number of other medicines. For example, Vitamin C appears in many biological processes and often appears as the correct nal answer. To remedy the bias problem, we propose to evaluate the model in both the original setting and the de-biased setting. In the de-biased setting, all proteins and medicines are replaced with a special token x$ITEM$y. The model is required to perform reasoning based on the context

of the proteins/medicines, rather than relying on the name of the proteins/medicines.

5.5 Experiments Results

5.5.1 Discover Reasoning Chains

HotpotQA. We first evaluate the model on the 2-hop HotpotQA task. Our best-performed model first selects the tail passage p_t and then the head passage p_h, because the number of candidates of the tail is smaller (2 per question). Table 5.1 shows the results. First, training a Ranker with distant supervision performs significantly better than the random baseline, showing that the training process itself has a certain degree of denoising ability to distinguish the more informative signals from distant supervision labels.

By introducing additional inductive bias of orders, the conditional selection model further improves with a large margin. Finally, our cooperative game gives the best performance, showing that a trained Reasoner can ignore entity links that are irrelevant to the reasoning chain.

Table 5.2 demonstrates the effect of selecting directions, together with the methods' recall on head passages and tail passages. The latter is evaluated on a subset of bridge-type questions in HotpotQA, which has no ambiguous support annotations in passage orders; that is, among the two human-labeled supporting passages, only one contains the answer and thus must be a tail. The results show that selecting tail first performs better. The cooperative game mainly improves the head selection.

MedHop. Results in Table 5.1 show that recovering chains from MedHop is a much harder task: first, the large number of distant supervision chains in C introduce too much noise so the Distant Supervised Ranker improves only 3%; second, the dependent model leads to no improvement because C is strictly ordered given our data construction. Our cooperative game manages to remain effective and gives further improvement.

Model	HotpotQA	MedHop
Random	40.3%	56.0%
Distant Supervised MatchLSTM	74.0%	59.3%
Conditional Selection	84.7%	59.3%
Cooperative Game	87.2%	62.6%

Table 5.1: Reasoning Chain selection results on HotpotQA.

Model - Hotpot	Head/Tail	EM
Conditional Selection (Head to Tail)	80.7/95.0%	77.1%
Conditional Selection (Tail to Head)	88.1/96.2%	84.7%
+ Cooperative Reasoner	90.1/96.7%	87.2%

Table 5.2: Ablation test on HotpotQA.

Models (Original)	Accuracy (%)
Random Guess	13.9
BiDAF (Seo et al., 2016)	47.8
FastQA (Weissenborn et al., 2017)	23.1
EPAr (Yichen Jiang and Bansal, 2019)	60.3
Ours	58.7
Ours (dropout = 0.1)	57.8
Ours (dropout = 0.2)	57.6

Table 5.3: Performance on MedHop original setting.

Models (Masked)	Accuracy (%)
Random Guess	13.9
BiDAF (Seo et al., 2016)	33.7
FastQA (Weissenborn et al., 2017)	31.3
EPAr (Yichen Jiang and Bansal, 2019)	41.6
Ours	48.5
Ours (dropout = 0.1)	50.0
Ours (dropout = 0.2)	49.7

Table 5.4: Performance on MedHop de-biased setting.

5.5.2 Medical Text Reasoning

We evaluate the medical text reasoning model on the Medhop dataset. In addition to the original challenge, we propose a de-biased setting, where the models have no access to the protein/medicine identi ers (and their entity embedding). The proteins/medicines are treated as placeholders and the model needs to focus on the context rather than the protein/medicine name to make a decision. The de-biased setting aims to alleviate the bias of the Medhop challenge, where a model can achieve high performance by selecting the most frequent answers that appeared during training. This bias emerges due to the fact that some medicines are highly active and interact with a wide range of medicines during training and testing.

Table 5.3 and Table 5.4 show the performance of di erent models on the Medhop dataset. In the original setting, our model performs signi cantly better than the popular baseline models on the MedHop leaderboard and is comparable to the current state-of-the-art model EPAr (Yichen Jiang and Bansal, 2019). In the de-biased setting, without knowing the protein/medicine names, the baseline models' performance drops signi cantly. Our model achieves state-of-the-art performance on the de-biased setting, exceeding EPAr by more than 7% in accuracy.

Experiments also show that in the original setting, applying dropout layers has little e ect on the nal accuracy, while in the de-biased setting, the dropout probability p 0:1 yields the best results.

5.6 Summary

5.6.1 Summary of Contributions

In this chapter, we proposed the problem of discovering reasoning chains in multi-hop QA with weakly-supervised signal, and we proposed a step-wise neuro-symbolic model for the

Medhop challenge, which is a text reasoning task in the medical domain.

For the task of discovering reasoning chains, we adopted a cooperative game approach where a Ranker and a Reasoner cooperate to select the most con dent reasoning chains. Experiments on the HotpotQA and MedHop benchmarks show the e ectiveness of the proposed approach.

For the task of medical text reasoning, our proposed model achieved competitive performance in the original setting and state-of-the-art performance in the de-biased setting.

5.6.2 Potential Extensions

The results of the Medhop challenge (Table 5.3 and Table 5.4) show that all models are vulnerable to the dataset biases that exist in the training set. To prevent the model from learning the dataset bias, one popular approach is to rst manually identify the source of the bias and then train a de-biased model that learns in the orthogonal space of the bias (Clark et al., 2019, 2020). Though the orthogonal methods are proven to be useful in many situations (Clark et al., 2019), they require a bias-only model as the prior information. Instead of identifying the biases in the dataset, it could be easier to rst identify the general logic of the task and then build a neuro-symbolic model with the identi ed task logic. As one of the potential future works, it is worthwhile to research how to build neuro-symbolic models that are less vulnerable to dataset biases.

Chapter 6

Conclusions

Neural network models have greatly advanced many areas of artificial intelligence research, achieving state-of-the-art performance on multiple challenging applications. However, recent research has revealed challenges of neural network models, which include but are not limited to the property of being a black-box model, the lack of the ability to generalize to out-of-distribution samples, and potential failures in compositional generalization. Moreover, it is generally hard to control the learning process of neural network models solely with back-propagation and gradient descent methods. With the help of logic frameworks and symbolic operators, neuro-symbolic models often provide a faithful explanation of their inner mechanism, which greatly helps the process of downstream application and diagnosis.

6.1 Summary of Contributions

In this thesis, we approach the aforementioned challenges by developing neuro-symbolic models. Instead of directly applying black-box neural network models to large annotated datasets, neural symbolic models incorporate various task-specific logic frameworks and logic operations into end-to-end neural network models, which have great impact on the way neural network models learn and perform reasoning over existing datasets.

As a novel contribution to neuro-symbolic research, this thesis focuses initially on the

task of natural language inference (NLI). We explore ways to build neural network models that follow a well-formulated natural logic theory (MacCartney, 2009), which is a rule-based symbolic framework developed to solve the NLI problem. In this thesis, we reveal several unique challenges for building an end-to-end natural logic model for NLI, and propose practical solutions to those challenges, presenting multiple neuro-symbolic models that simultaneously achieve competitive performance on both the standard benchmark SNLI (Bowman et al., 2015) and various linguistic probes (Yanaka et al., 2019a,b, 2020). Meanwhile, our methods provide faithful and easy-to-interpret explanations of their inner mechanism.

Though natural logic is designed exclusively for NLI, the ideas of step-wise reasoning and surface-form reasoning have a profound impact on other artificial intelligence applications. Instead of relying on logic-form-based task-specific language and operations, which generalize poorly given new tasks and problems, surface form-reasoning assumes that all reasoning processes happen on the language syntax and semantic level, which is to some extent universal to all the natural language processing problems.

Based on this motivation, we explore how to extend the proposed model and the interpretation method to multi-hop QA. In this thesis, we proposed a model that accurately locates from a significant amount of text useful pieces of evidence, and the model can be trained without direct supervision. Then we built a neuro-symbolic model that achieved state-of-the-art performance in detecting medicine-to-medicine interactions from medical text.

6.2 Future Research

Following the current research progress, multiple research topics are worthy of further exploration.

Spurious Reasoning Problem. The spurious reasoning problem, which is described in Chapter 3 and Chapter 4, greatly affects the quality of the explanations generated by the neuro-symbolic models. The problem happens when the model is trained with weak supervision: without extra supervision for the intermediate logic steps, it is challenging for a neuro-symbolic model to learn the correct symbolic programs during end-to-end training. In this thesis, we propose to use lexical constraints (Chapter 3) or introspective revision (Chapter 4) to mitigate the spurious reasoning problem, and in Chapter 4, we also show that the uni-directional attention model GPT-2 may be less affected by the spurious reasoning problem compared to the bi-directional model BERT. It is worthwhile to investigate in future research how to prevent the entangling problems in transformer-based neuro-symbolic models.

Natural Logic Variants and Their Applications. In this thesis, we show that natural logic theory can be incorporated into neural network models to enhance the model's ability in logic reasoning and interpretability. It is also shown that, though being a sound logic framework, there is opportunity and difficulty in applying natural logic to applications other than NLI. It is worthwhile to research how to adapt natural logic to various natural language reasoning applications, where the model can achieve good performance and interpretability with the help of step-wise, surface-form reasoning.

Candidate reasoning applications include:

- Multi-modal question answering. It will be interesting to investigate whether the step-wise reasoning in natural logic can be applied to visual question answering, where the image serves as the premise and the text statement as the hypothesis.
- Fact verification. Diverse fact verification tasks, such as fake news detection and table-based fact verification (Chen et al., 2019c), can be the next playground for natural logic reasoning.

Few-Shot Natural Logic Reasoning. Investigating few-shot behaviors of the natural logic reasoning model is a promising research direction.

In this thesis, we assume that the ground-truth alignment and semantic relations between components of the premise and hypothesis are not available, due to the high expense of further annotation. However, with a small number of extra annotations, we can design experiments to see whether they can alleviate the spurious reasoning problem and improve the accuracy of intermediate relations.

Recently, prompting method (Brown et al., 2020b; Liu et al., 2021), in-context learning, and the chain-of-thought method (Wei et al., 2022) illuminated an unique opportunity for few-shot natural logic reasoning and NLI. Wei et al. (2022) showed that GPT-3 (Brown et al., 2020b) and PALM (Chowdhery et al., 2022) can generate natural language sentences that solve complex natural language reasoning problems with a few carefully designed chain-of-thought prompts. It is worth a try in the future work where we design natural logic based chain of thought and use GPT-3 or PALM to generate natural-logic-based reasoning steps and explanations.

Bibliography

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*.

Saeed Amizadeh, Hamid Palangi, Alex Polozov, Yichen Huang, and Kazuhito Koishida. 2020a. Neuro-symbolic visual reasoning: Disentangling. In *International Conference on Machine Learning*, pages 279{290. PMLR.

Saeed Amizadeh, Hamid Palangi, Oleksandr Polozov, Yichen Huang, and Kazuhito Koishida. 2020b. Neuro-symbolic visual reasoning: Disentangling" visual" from" reasoning". *arXiv preprint arXiv:2006.11524*, 2.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016a. Neural module networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 39{48.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016b. Neural module networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, United States.

Gabor Angeli and Christopher D Manning. 2014. Naturalli: Natural logic inference for common sense reasoning. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, Doha, Qatar.

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344{354.

Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. 2019. Learning to retrieve reasoning paths over wikipedia graph for question answering. *arXiv preprint arXiv:1911.10470*.

Sebastian Bach, Alexander Binder, Gregoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classi er decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140.

Johan van Benthem. 1988. The semantics of variety in categorial grammar. *Categorial grammar*, 25:37{55.

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2009. The fth pascal recognizing textual entailment challenge. In *TAC*.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Lisbon, Portugal.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla

Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020a. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877{1901.

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

Oana-Maria Camburu, Tim Rocktaschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. In *Advances in Neural Information Processing Systems*.

Shiyu Chang, Yang Zhang, Mo Yu, and Tommi Jaakkola. 2020. Invariant rationalization. In *International Conference on Machine Learning*, pages 1448{1458. PMLR.

Jifan Chen, Shih-ting Lin, and Greg Durrett. 2019a. Multi-hop question answering via reasoning chains. *arXiv preprint arXiv:1910.02610*.

Kunlong Chen, Weidi Xu, Xingyi Cheng, Zou Xiaochuan, Yuyu Zhang, Le Song, Taifeng Wang, Yuan Qi, and Wei Chu. 2020a. Question directed graph attention network for numerical reasoning over text. *arXiv preprint arXiv:2009.07448*.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Diana Inkpen, and Si Wei. 2017a. Neural natural language inference models enhanced with external knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, Melbourne, Australia.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017b.

Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, Vancouver, Canada.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2016. Enhanced lstm for natural language inference. *arXiv preprint arXiv:1609.06038*.

Qianglong Chen, Feng Ji, Xiangji Zeng, Feng-Lin Li, Ji Zhang, Haiqing Chen, and Yin Zhang. 2021a. Kace: Generating knowledge aware contrastive explanations for natural language inference. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2516{2527.

Tongfei Chen, Zhengping Jiang, Adam Poliak, Keisuke Sakaguchi, and Benjamin Van Durme. 2019b. Uncertain natural language inference. *arXiv preprint arXiv:1909.03042*.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2019c. Tabfact: A large-scale dataset for table-based fact veri cation. In *International Conference on Learning Representations*.

Xinyun Chen, Chen Liang, Adams Wei Yu, Denny Zhou, Dawn Song, and Quoc V Le. 2019d. Neural symbolic reader: Scalable integration of distributed and symbolic representations for reading comprehension. In *International Conference on Learning Representations*.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020b. Uniter: Universal image-text representation learning. In *ECCV*.

Zeming Chen, Qiyue Gao, and Lawrence S Moss. 2021b. Neurallog: Natural language inference with joint neural and logical reasoning. *arXiv preprint arXiv:2105.14167*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2019. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. 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 4069{4082.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2020. Learning to model and ignore dataset bias with mixed capacity ensembles. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3031{3045.

Alexis Conneau, German Kruszewski, Guillaume Lample, Lo•c Barrault, and Marco Baroni. 2018. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. *arXiv preprint arXiv:1805.01070*.

Robin Cooper, Dick Crouch, Jan van Eijck, Chris Fox, Josef van Genabith, Jan Jaspars, Hans Kamp, David Milward, Manfred Pinkal, Massimo Poesio, et al. 1996. Using the framework.

Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alex Smola, and Andrew McCallum. 2018. Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. In *Proceedings of ICLR 2018*.

Luc De Raedt, Robin Manhaeve, Sebastijan Dumancic, Thomas Demeester, and Angelika

Kimmig. 2019. Neuro-symbolic= neural+ logical+ probabilistic. In *NeSy'19@ IJCAI, the 14th International Workshop on Neural-Symbolic Learning and Reasoning*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT 2019*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, Minneapolis, USA.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C Wallace. 2020. Eraser: A benchmark to evaluate rationalized nlp models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443{4458.

Ming Ding, Chang Zhou, Qibin Chen, Hongxia Yang, and Jie Tang. 2019. Cognitive graph for multi-hop reading comprehension at scale. In *Proceedings of ACL 2019*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. *arXiv preprint arXiv:1903.00161*.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2017. Hot ip: White-box adversarial examples for text classi cation. *arXiv preprint arXiv:1712.06751*.

Yuwei Fang, Siqi Sun, Zhe Gan, Rohit Pillai, Shuohang Wang, and Jingjing Liu. 2020. Hierarchical graph network for multi-hop question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8823{ 8838.

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. Pathologies of neural models make interpretations di cult. *arXiv preprint arXiv:1804.07781*.

Yufei Feng, Xiaoyu Yang, Xiaodan Zhu, and Michael Greenspan. 2022. Neuro-symbolic natural logic with introspective revision for natural language inference. *Transactions of the Association for Computational Linguistics*, 10:240{256.

Yufei Feng, Mo Yu, Wenhan Xiong, Xiaoxiao Guo, Junjie Huang, Shiyu Chang, Murray Campbell, Michael Greenspan, and Xiaodan Zhu. 2021. Learning to recover reasoning chains for multi-hop question answering via cooperative games. *The 34th Canadian Conference on Arti cial Intelligence (Canadian AI 2021)*.

Yufei Feng, Zi'ou Zheng, Quan Liu, Michael Greenspan, and Xiaodan Zhu. 2020. Exploring end-to-end di erentiable natural logic modeling. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1172{1185.

Herve Gallaire and Jack Minker. 1978. Logic and data bases, symposium on logic and data bases, centre d'etudes et de recherches de toulouse, 1977. *Advances in Data Base Theory*.

Zhe Gan, Yen-Chun Chen, Linjie Li, Chen Zhu, Yu Cheng, and Jingjing Liu. 2020. Large-scale adversarial training for vision-and-language representation learning. *Advances in Neural Information Processing Systems*, 33:6616{6628.

Artur d'Avila Garcez, Tarek R Besold, Luc De Raedt, Peter Foldiak, Pascal Hitzler, Thomas Icard, Kai-Uwe Kuhnberger, Luis C Lamb, Risto Miikkulainen, and Daniel L Silver. 2015. Neural-symbolic learning and reasoning: contributions and challenges. In *2015 AAAI Spring Symposium Series*.

Artur d'Avila Garcez, Marco Gori, Luis C Lamb, Luciano Sera ni, Michael Spranger, and Son N Tran. 2019. Neural-symbolic computing: An e ective methodology for principled integration of machine learning and reasoning. *arXiv preprint arXiv:1905.06088*.

Artur S d'Avila Garcez, Krysia Broda, Dov M Gabbay, et al. 2002. *Neural-symbolic learning systems: foundations and applications*. Springer Science & Business Media.

Artur SD'Avila Garcez, Luis C Lamb, and Dov M Gabbay. 2008. *Neural-symbolic cognitive reasoning*. Springer Science & Business Media.

Matt Gardner, Yoav Artzi, Victoria Basmova, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, et al. 2020. Evaluating models' local decision boundaries via contrast sets. *arXiv preprint arXiv:2004.02709*.

Atticus Geiger, Kyle Richardson, and Christopher Potts. 2020. Neural natural language inference models partially embed theories of lexical entailment and negation. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 163{173.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking nli systems with sentences that require simple lexical inferences. *arXiv preprint arXiv:1805.02266*.

Nitish Gupta, Kevin Lin, Dan Roth, Sameer Singh, and Matt Gardner. 2019a. Neural module networks for reasoning over text. *arXiv preprint arXiv:1912.04971*.

Nitish Gupta, Kevin Lin, Dan Roth, Sameer Singh, and Matt Gardner. 2019b. Neural module networks for reasoning over text. *arXiv preprint arXiv:1912.04971*.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R Bowman, and Noah A Smith. 2018. Annotation artifacts in natural language inference data. In *NAACL-HLT (2)*.

Kelvin Guu, Panupong Pasupat, Evan Zheran Liu, and Percy Liang. 2017. From language to programs: Bridging reinforcement learning and maximum marginal likelihood. *arXiv preprint arXiv:1704.07926*.

Xiaochuang Han, Byron C Wallace, and Yulia Tsvetkov. 2020. Explaining black box predictions and unveiling data artifacts through in uence functions. *arXiv preprint arXiv:2005.06676*.

Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. 2009. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

Geo rey E Hinton. 1984. Distributed representations.

Matthew Honnibal, Ines Montani, So e Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Md Mosharaf Hossain, Venelin Kovatchev, Pranoy Dutta, Ti any Kao, Elizabeth Wei, and Eduardo Blanco. 2020. An analysis of natural language inference benchmarks through the lens of negation. In *EMNLP*, pages 9106{9118.

Hai Hu, Qi Chen, Kyle Richardson, Atreyee Mukherjee, Lawrence S Moss, and Sandra Kuebler. 2020. MonaLog: a lightweight system for natural language inference based on monotonicity. In *Proceedings of the Society for Computation in Linguistics 2020*.

Thomas F Icard. 2012. Inclusion and exclusion in natural language. *Studia Logica*.

Thomas F Icard and Lawrence S Moss. 2014. Recent progress on monotonicity. In *Linguistic Issues in Language Technology*. Citeseer.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. *arXiv preprint arXiv:1804.06059*.

Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable NLP systems: How should we de ne and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198{4205, Online. Association for Computational Linguistics.

Sarthak Jain, Sarah Wiegre e, Yuval Pinter, and Byron C Wallace. 2020. Learning to faithfully rationalize by construction. *arXiv preprint arXiv:2005.00115*.

Zhongtao Jiang, Yuanzhe Zhang, Zhao Yang, Jun Zhao, and Kang Liu. 2021. Alignment rationale for natural language inference. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5372{5387.

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.

Aikaterini-Lida Kalouli, Richard Crouch, and Valeria de Paiva. 2020. Hy-nli: a hybrid system for natural language inference. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5235{5249.

J.J. Katz. 1972. *Semantic Theory*. Harper & Row.

Tushar Khot, Daniel Khashabi, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2020. Text modular networks: Learning to decompose tasks in the language of existing models. *arXiv preprint arXiv:2009.00751*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, San Diego, CA.

Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via in uence functions. In *International conference on machine learning*, pages 1885{1894. PMLR.

Keshav Kolluru, Samarth Aggarwal, Vipul Rathore, Soumen Chakrabarti, et al. 2020. Imojie: Iterative memory-based joint open information extraction. *arXiv preprint arXiv:2005.08178*.

Yuta Koreeda and Christopher D Manning. 2021. Contractnli: A dataset for document-level natural language inference for contracts. *arXiv preprint arXiv:2110.01799*.

Sawan Kumar and Partha Talukdar. 2020. Nile: Natural language inference with faithful natural language explanations. *arXiv preprint arXiv:2005.12116*.

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.

George Lako . 1970. Linguistics and natural logic. *Synthese*.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107{117.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*.

Qing Li, Siyuan Huang, Yining Hong, Yixin Chen, Ying Nian Wu, and Song-Chun Zhu. 2020. Closed loop neural-symbolic learning via integrating neural perception, grammar parsing, and symbolic reasoning. In *International Conference on Machine Learning*, pages 5884{5894. PMLR.

Hugo Liu and Push Singh. 2004. Conceptnet | a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211{226.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

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

Scott M Lundberg and Su-In Lee. 2017. A uni ed approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

Bill MacCartney. 2009. *Natural Language Inference*. Ph.D. thesis, Stanford University.

Bill MacCartney, Michel Galley, and Christopher D Manning. 2008. A phrase-based alignment model for natural language inference. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 802{811.

Bill MacCartney and Christopher D. Manning. 2008. Modeling semantic containment and exclusion in natural language inference. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 521{528, Manchester, UK. Coling 2008 Organizing Committee.

Bill MacCartney and Christopher D. Manning. 2009. An extended model of natural logic. In *Proceedings of the Eight International Conference on Computational Semantics*, pages 140{156, Tilburg, The Netherlands. Association for Computational Linguistics.

Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. 2018. Deepproblog: Neural probabilistic logic programming. *Advances in Neural Information Processing Systems*, 31:3749{3759.

Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B Tenenbaum, and Jiajun Wu. 2019. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. *arXiv preprint arXiv:1904.12584*.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Ra aella Bernardi, Roberto Zamparelli, et al. 2014. A sick cure for the evaluation of compositional distributional semantic models. In *Lrec*, pages 216{223. Reykjavik.

R Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *arXiv preprint arXiv:1902.01007*.

George A Miller. 1998. *WordNet: An electronic lexical database*. MIT press.

Sewon Min, Danqi Chen, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019a. A discrete hard em approach for weakly supervised question answering. In *Proceedings of EMNLP 2019*.

Sewon Min, Victor Zhong, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019b. Multi-hop reading comprehension through question decomposition and rescoring. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6097{ 6109.

Pasquale Minervini, Matko Bosnjak, Tim Rocktaschel, Edward Grefenstette, and Sebastian Riedel. 2018. Scalable neural theorem proving on knowledge bases and natural language.

Pasquale Minervini, Sebastian Riedel, Pontus Stenetorp, Edward Grefenstette, and Tim Rocktaschel. 2020. Learning reasoning strategies in end-to-end di erentiable proving. In *International Conference on Machine Learning*, pages 6938{6949. PMLR.

Anshuman Mishra, Dhruvesh Patel, Aparna Vijayakumar, Xiang Lorraine Li, Pavan Kapanipathi, and Kartik Talamadupula. 2021. Looking beyond sentence-level natural language inference for question answering and text summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1322{1336.

Rowan Nairn, Cleo Condoravdi, and Lauri Karttunen. 2006. Computing relative polarity for textual inference. In *Proceedings of the fth international workshop on inference in computational semantics (icos-5)*.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gucehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280{290, Berlin, Germany. Association for Computational Linguistics.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2019. Adversarial nli: A new benchmark for natural language understanding. *arXiv preprint arXiv:1910.14599*.

Yixin Nie, Xiang Zhou, and Mohit Bansal. 2020. What can we learn from collective human opinions on natural language inference data? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Kosuke Nishida, Kyosuke Nishida, Masaaki Nagata, Atsushi Otsuka, Itsumi Saito, Hisako Asano, and Junji Tomita. 2019. Answering while summarizing: Multi-task learning for multi-hop qa with evidence extraction. *arXiv preprint arXiv:1905.08511*.

Jessica Ouyang and Kathy McKeown. 2019. Neural network alignment for sentential paraphrases. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy. Association for Computational Linguistics.

Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. *arXiv preprint arXiv:1606.01933*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014a. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014b. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, Doha, Qatar.

Ethan Perez, Siddharth Karamcheti, Rob Fergus, Jason Weston, Douwe Kiela, and

Kyunghyun Cho. 2019. Finding generalizable evidence by learning to convince q&a models. In *Proceedings of EMNLP 2019*.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. *arXiv preprint arXiv:1805.01042*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748{8763. PMLR.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Je rey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why should I trust you?: Explaining the predictions of any classi er. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135{1144. ACM.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging nlp models. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of nlp models with checklist. *arXiv preprint arXiv:2005.04118*.

Kyle Richardson, Hai Hu, Lawrence S Moss, and Ashish Sabharwal. 2020. Probing natural language inference models through semantic fragments. In *Proceedings of the Thirty-Fourth AAAI Conference on Arti cial Intelligence (AAAI)*, New York, USA.

Tim Rocktaschel and Sebastian Riedel. 2017. End-to-end di erentiable proving. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, Long Beach, USA.

Swarnadeep Saha, Yixin Nie, and Mohit Bansal. 2020. Conjnli: Natural language inference over conjunctive sentences. *arXiv preprint arXiv:2010.10418*.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618{626.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention ow for machine comprehension. *arXiv preprint arXiv:1611.01603*.

So a Serrano and Noah A Smith. 2019. Is attention interpretable? *arXiv preprint arXiv:1906.03731*.

Xing Shi, Inkit Padhi, and Kevin Knight. 2016. Does string-based neural mt learn source syntax? In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 1526{1534.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation di erences. In *International conference on machine learning*, pages 3145{3153. PMLR.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classi cation models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Koustuv Sinha, Prasanna Parthasarathi, Joelle Pineau, and Adina Williams. 2020. Unnatural language inference. *arXiv preprint arXiv:2101.00010*.

Xuelin Situ, Ingrid Zukerman, Cecile Paris, Sameen Maruf, and Gholamreza Ha ari. 2021. Learning to explain: Generating stable explanations fast. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viegas, and Martin Wattenberg. 2017. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631{1642.

Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018. Supervised open information extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 885{895, New Orleans, Louisiana. Association for Computational Linguistics.

Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2019. Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*.

Sanjay Subramanian, Ben Bogin, Nitish Gupta, Tomer Wolfson, Sameer Singh, Jonathan

Berant, and Matt Gardner. 2020. Obtaining faithful interpretations from compositional neural networks. *arXiv preprint arXiv:2005.00724*.

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in Neural Information Processing Systems*, Montreal Canada.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3319{3328. JMLR. org.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. *arXiv preprint arXiv:1803.06643*.

Alon Talmor, Oyvind Tafjord, Peter Clark, Yoav Goldberg, and Jonathan Berant. 2020. Leap-of-thought: Teaching pre-trained models to systematically reason over implicit knowledge. *Advances in Neural Information Processing Systems*, 33.

Shawn Tan, Yikang Shen, Chin-wei Huang, and Aaron Courville. 2019. Investigating biases in textual entailment datasets. *arXiv preprint arXiv:1906.09635*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and veri cation. *arXiv preprint arXiv:1803.05355*.

Ming Tu, Kevin Huang, Guangtao Wang, Jing Huang, Xiaodong He, and Bowen Zhou. 2020. Select, answer and explain: Interpretable multi-hop reading comprehension over multiple documents. In *Proceedings of the AAAI Conference on Arti cial Intelligence*, volume 34, pages 9073{9080.

V ctor Manuel Sanchez Valencia. 1991. *Studies on natural logic and categorial grammar*. Universiteit van Amsterdam.

Johan Van Benthem. 1986. *Essays in logical semantics*. Springer.

Johan Van Benthem. 1995. *Language in Action: categories, lambdas and dynamic logic*. MIT Press.

Johan Van Benthem et al. 2008. *A brief history of natural logic*. London College Publications.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Eric Wallace, Matt Gardner, and Sameer Singh. 2020. Interpreting predictions of NLP models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, pages 20{23.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018a. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Haoyu Wang, Mo Yu, Xiaoxiao Guo, Rajarshi Das, Wenhan Xiong, and Tian Gao. 2019. Do multi-hop readers dream of reasoning chains? *arXiv preprint arXiv:1910.14520*.

Shuohang Wang and Jing Jiang. 2016. Learning natural language inference with lstm. In *Proceedings of NAACL-HLT 2016*, pages 1442{1451.

Shuohang Wang, Mo Yu, Xiaoxiao Guo, Zhiguo Wang, Tim Klinger, Wei Zhang, Shiyu Chang, Gerald Tesauro, Bowen Zhou, and Jing Jiang. 2018b. R3: Reinforced ranker-reader for open-domain question answering. In *Proceedings of AAAI 2018*.

Leon Weber, Pasquale Minervini, Jannes Münchmeyer, Ulf Leser, and Tim Rocktäschel. 2019. Nlprolog: Reasoning with weak unification for question answering in natural language. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, Austin, Texas, United States.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Dirk Weissenborn, Georg Wiese, and Laura Seiffe. 2017. Fastqa: A simple and efficient neural architecture for question answering. *ArXiv*, abs/1703.04816.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287{302.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory networks. *arXiv preprint arXiv:1410.3916*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112{1122.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229{256.

Yunxuan Xiao, Yanru Qu, Lin Qiu, Hao Zhou, Lei Li, Weinan Zhang, and Yong Yu. 2019. Dynamically fused graph network for multi-hop reasoning. In *Proceedings of ACL 2019*.

Wenhan Xiong, Thien Hoang, and William Yang Wang. 2017. Deeppath: A reinforcement learning method for knowledge graph reasoning. In *Proceedings of EMNLP 2017*.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048{2057. PMLR.

Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, and Kentaro Inui. 2020. Do neural models learn systematicity of monotonicity inference in natural language? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6105{6117.

Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, Kentaro Inui, Satoshi Sekine, Lasha Abzianidze, and Johan Bos. 2019a. Can neural networks understand monotonicity reasoning? In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Austin, Texas, United States.

Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, Kentaro Inui, Satoshi Sekine, Lasha Abzianidze, and Johan Bos. 2019b. Help: A dataset for identifying shortcomings of neural models in monotonicity reasoning. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM)*, Minneapolis, Minnesota, USA.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of EMNLP 2018*.

Xuchen Yao, Benjamin Van Durme, Chris Callison-Burch, and Peter Clark. 2013. A

lightweight and high performance monolingual word aligner. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 702{707.

Chih-Kuan Yeh, Joon Kim, Ian En-Hsu Yen, and Pradeep K Ravikumar. 2018. Representer point selection for explaining deep neural networks. *Advances in neural information processing systems*, 31.

Yen-chun Chen Yichen Jiang, Nitish Joshi and Mohit Bansal. 2019. Explore, propose, and assemble: An interpretable model for multi-hop reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Mo Yu, Yang Zhang, Shiyu Chang, and Tommi Jaakkola. 2021. Understanding interlocking dynamics of cooperative rationalization. *Advances in Neural Information Processing Systems*, 34.

Yijia Zhang, Qingyu Chen, Zhihao Yang, Hongfei Lin, and Zhiyong Lu. 2019. Biowordvec, improving biomedical word embeddings with subword information and mesh. *Scienti c data*, 6(1):52.

Yimeng Zhuang and Huadong Wang. 2019. Token-level dynamic self-attention network for multi-passage reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2252{2262.