DIFFERENTIABLE CONSTRAINT-BASED SOLVERS FOR EXPLANATION-BASED MULTI-HOP INFERENCE

A THESIS SUBMITTED TO THE UNIVERSITY OF MANCHESTER
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
IN THE FACULTY OF ENGINEERING AND PHYSICAL SCIENCES

2022

By
Mokanarangan Thayaparan
School of Computer Science

Contents

Word Count: 25,760

List of Tables

List of Figures

Abstract

DIFFERENTIABLE CONSTRAINT-BASED SOLVERS FOR
EXPLANATION-BASED MULTI-HOP INFERENCE
Mokanarangan Thayaparan
A thesis submitted to the University of Manchester
for the degree of Doctor of Philosophy, 2022

Explanation-based Question Answering (XQA) for complex questions involving scientific and common-sense reasoning is often modelled as a *multi-hop* reasoning. Constrained optimization solvers based on Integer Linear Programming (ILP) have been proposed to address these multi-hop inference tasks. This family of approaches provides a viable mechanism to encode explicit and controllable assumptions, casting multi-hop as an optimal subgraph selection problem.

However, these approaches have shown diminishing returns with an increasing number of hops suffering from a phenomenon called *semantic drift*. Moreover, these approaches are typically *non-differentiable* and cannot be integrated as part of a deep neural network. This shortcoming prevents these methods from learning end-to-end on annotated corpora and achieving performance comparable to deep learning counterparts.

This thesis attempts to solve these problems by presenting the following contributions:

- Introduce a novel model (*ExplantionLP*) that performs inference encoding grounding-abstract chains for explanation-based multi-hop inference and reduces *semantic drift*. We demonstrate that *ExplantionLP* is more robust to semantic drift when compared with graph-based and transformer-based approaches.

- Present the first hybrid model (*Diff*-Explainer) that integrates constrained optimization as part of a deep neural network via differentiable convex optimization,

allowing the fine-tuning of pre-trained transformers for downstream explanation-based multi-hop Inference task. We empirically demonstrate on scientific and common-sense QA benchmarks that integrating explicit constraints in an end-to-end differentiable framework can significantly improve the performance of non-differentiable ILP solvers.

- Propose a novel hybrid model (*Diff*-Comb Explainer) that integrates constrained optimization as part of a deep neural network via Differentiable BlackBox Combinatorial solvers, allowing the fine-tuning of pre-trained transformers for downstream explanation-based multi-hop Inference task. *Diff*-Comb Explainer demonstrates improved answer and explanation selection accuracy over non-differentiable solvers, transformers and existing differentiable constraint-based multi-hop inference frameworks.

We also present a systematic review of the explainable natural language inference field. In this survey, we present an analysis of existing benchmarks and models. Additionally, identifying the emerging research trends and highlighting challenges and opportunities for future work.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright

Acknowledgements

Firstly, I would like to thank my supervisor Andre Freitas for giving me this wonderful opportunity and guiding me through the challenging process. I drew a lot of confidence and strength from your belief in me.

I would like to thank my examiners, Prof. Chris Biemann and Prof. Angelo Cangelosi. for reviewing my thesis and engaging in a deep discussion during my viva.

I was lucky enough to share my PhD with wonderful people. I would like to thank Deborah and Alber for providing space in their home and sharing the warmth of their hearts. I would also like to thank Marco for being my partner-in-crime and for sharing his knowledge with me. I want to thank Jake and Julia for showing that life can be fun and research is not the only thing there is. The most cherished memories of my PhD were formed around you all.

I would like to thank all the people from the Reasoning & Explainable AI lab and the Department of Computer Science I had the opportunity to meet and work with, including but not limited to Mauricio, Conor, Hanadi, MingYang, Viktor, Oskar, Salim, Danilo, Guy and David. I also want to thank all the teachers who shaped me through these years.

My wife, Subanki, is the lighthouse and anchor of my life. It must have been challenging for her to let me leave and go through a phase of a long-distance relationship. But she supported me during these trying times with love and care from far shores. I love her with my entire heart.

Two people are the principal characters of this journey. The first would be my *Amma* (mother), who showed me courage and resilience come in all forms and shapes. She believed in me even when I did not and kept pushing me to new heights. Nothing of this would have been possible without her. The other person is my *Appa* (father). I lost my father during the final year of my PhD. It will always be painful for me that he could not see me finish my Ph.D. This achievement was my father's as much as mine. He ensured that I got the best possible education, a luxury where I come from. I hope he knew he was loved and built the path for me to walk.

Chapter 1

Introduction

1.1 Motivation

Explanation-based inference is the design of models capable of performing transparent inference through the generation of an *explanation* for the prediction. Explanation-based Question Answering (XQA) for complex questions involving scientific and commonsense reasoning is often modelled as a *multi-hop* reasoning problem (Jansen, 2018; Jansen et al., 2016). The goal of a typical XQA solver is to answer a given question and construct an explanation as a graph composed of multiple interconnected sentences (i.e., hops) supporting the answer (Jansen, 2018; Khashabi, Khot, Sabharwal, & Roth, 2018; Kundu et al., 2019).

For example, consider the question and answer presented in Figure 1.1. In order to answer this question, an XQA solver should be able to combine multiple facts from how a stick is an object and rubbing two objects together creates friction leading to heat being produced.

Recently, large-scale benchmarks have been proposed to train and evaluate models with multi-hop reasoning capabilities. These benchmarks covers a diverse set of reasoning domains including *open-domain* (Z. Yang et al., 2018), *scientific* (Clark et al., 2018; Khot et al., 2020), *commonsense* (Talmor et al., 2019) and format: *multiple choice selection* (Clark et al., 2018), *textual entailment* (Williams et al., 2017) and *span selection* (Z. Yang et al., 2018).

The current state-of-the-art (SOTA) models trained on these benchmarks for multi-hop inference are exclusively represented by Transformer-based models (He et al., 2020; Khashabi et al., 2020; Yadav et al., 2020). Pre-trained transformer models have been shown to learn natural language representations from large volumes of text data and

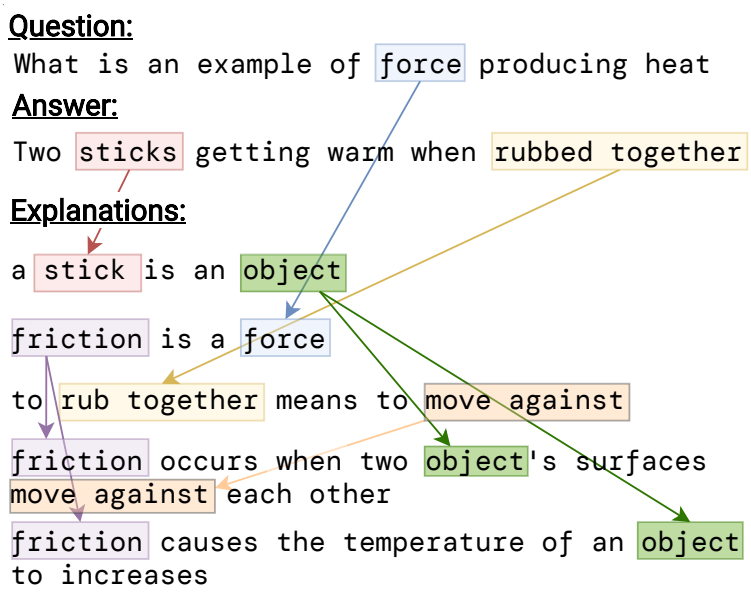

Figure 1.1: An example for explanation-based multi-hop inference (Jansen et al., 2018).

transfer this knowledge to downstream tasks like question answering, textual entailment, and language generation with little fine-tuning (Devlin et al., 2019; Y. Liu et al., 2019). However, Transformers are typically regarded as black-boxes (Liang et al., 2021), posing concerns about the interpretability and transparency of their predictions (Guidotti et al., 2018; Rudin, 2019).

Despite yielding high performance across various benchmarks, these SOTA deep learning models have been shown to exploit biases in the data (Gururangan et al., 2018; McCoy et al., 2019). In contrast, an explanation-based solver can provide an additional mechanism to investigate and analyze the internal reasoning mechanisms (Dua et al., 2020; Inoue et al., 2020; Ross et al., 2017). By focusing on explicit reasoning methods, research in explanation-based inference can lead to the development of novel models able to perform compositional generalization (Andreas et al., 2016a; N. Gupta et al., 2020) and discover abstract inference patterns in the data (Khot et al., 2020; Rajani et al., 2019), favouring few-shot learning and cross-domain transportability (Camburu et al., 2018).

1.2 Constrained Optimization for Multi-hop Inference

In this context, constrained optimization solvers based on Integer Linear Programming (ILP) have been proposed to address these multi-hop inference tasks (Khashabi et al.,

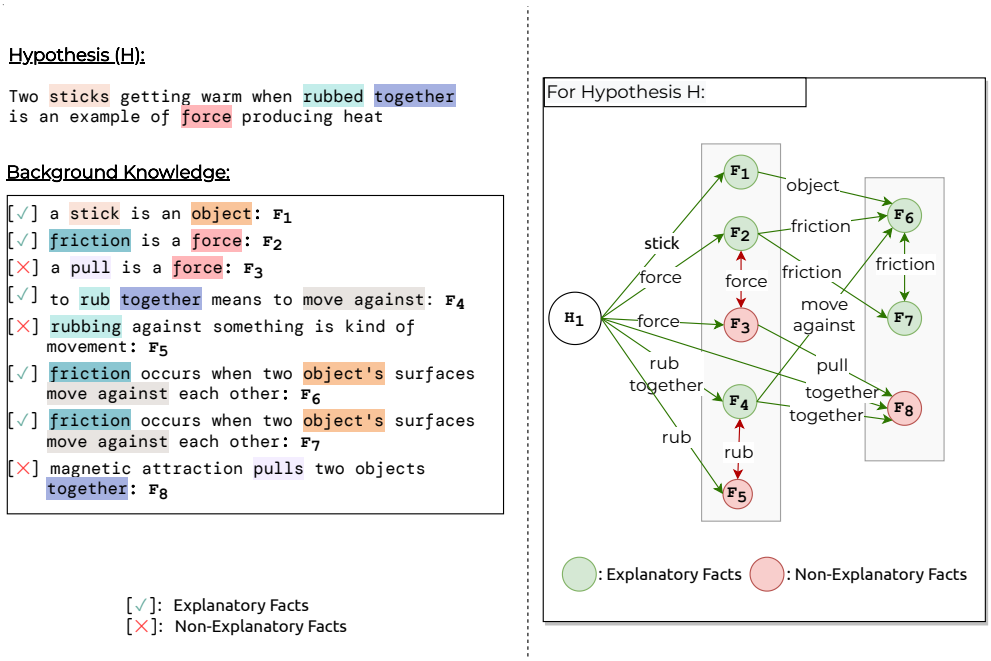

Figure 1.2: An example where relevant information needs to be extracted while discarding spurious facts to answer the question.

2016; Khashabi, Khot, Sabharwal, & Roth, 2018; Khot et al., 2017). This family of approaches provides a viable mechanism to encode explicit and controllable assumptions, casting multi-hop and explanation-based QA as an optimal subgraph selection problem (Clark et al., 2018; Jansen et al., 2018; Xie et al., 2020).

For example, refer to Figure 1.2, here an explanation-based solver should be able to identify that the central concept is of *friction* and its properties. In order to achieve this, the solver needs to be able abstract from *stick* to *object*, *friction* to *force* to connect with the facts about *friction*. The solver should also be able to filter out distracting knowledge about *pull* and *magnetic attraction* that has strong lexical overlaps but is not relevant to the hypothesis. This process can be cast as an optimal subgraph problem where a constraint-based solver aims to identify a subgraph of facts supporting the answer.

TableILP (Khashabi et al., 2016) is one of the earliest approaches to formulate the construction of explanations as an optimal sub-graph selection problem over a set of structured tables and evaluated on multiple-choice elementary science question answering. In contrast to TableILP, TupleILP (Khot et al., 2017) was able to perform inference over free-form text by building semi-structured representations using Open Information Extraction. SemanticILP (Khashabi, Khot, Sabharwal, & Roth, 2018) also

comes from the same family of solvers that leveraged different semantic abstractions, including semantic role labelling, named entity recognition and lexical chunkers for inference.

1.3 Problem Formulation

In this thesis, we focus on multiple-choice science question answering to evaluate the capabilities of constraint-based solvers. The motivation to choose science questions is because, unlike open-domain factoid-based question answering, it requires complex forms of inference, including causal, model-based and example-based reasoning (Clark et al., 2018; Clark et al., 2013; Jansen, 2018; Jansen et al., 2016). Our goal is also aided by the availability of explanations supporting benchmarks for science questions (Jansen et al., 2018; Xie et al., 2020).

The problem of Explanation-based Multi-Hop Question Answering (XQA) can be stated as follows:

Definition 1 (*Explanations in Multi-Hop Question Answering*)**.** Given a question Q, answer a and a knowledge base F_{kb} (composed of natural language sentences), we say that we may *infer* hypothesis h (where hypotheses h is the concatenation of Q with a) if there exists a subset (F_{exp}) of supporting facts $\{f_1, f_2, \ldots\} \subseteq F_{kb}$ of statements which would allow to arrive at h from $\{f_1, f_2, \ldots\}$. We call this set of facts an *explanation* for h.

Given a question (Q) and a set of candidate answers $C = \{c_1,\ c_2,\ c_3, ...,\ c_n\}$ ILP-based approaches (Khashabi et al., 2016; Khot et al., 2017) convert them into a list of hypothesis $H = \{h_1,\ h_2,\ h_3, \ldots,\ h_n\}$ by concatenating question and candidate answer. For each hypothesis h_i these approaches typically adopt a retrieval model (e.g: TF-IDF, BM25 (Robertson, Zaragoza, et al., 2009)), to select a list of candidate explanatory facts $F = \{f_1,\ f_2,\ f_3,\ \ldots,\ f_k\}$, and construct a weighted graph $G = (V, E, W)$ with edge weights $W : E \rightarrow \mathbb{R}$ where $V = \{\{h_i\} \cup F\}$, edge weight W_{ik} of each edge E_{ik} denote how relevant a fact f_k is with respect to the hypothesis h_i.

Based on these definitions, ILP-based QA can be defined as follows:

Definition 2 (*ILP-Based Multi-Hop QA*)**.** Find a subset $\tilde{V} \subseteq V$, $h \in \tilde{V}$, $\tilde{V} \setminus \{h\} = F_{exp}$ and $\tilde{E} \subseteq E$ such that the induced subgraph $\tilde{G} = (\tilde{V}, \tilde{E})$ is connected, weight $W[\tilde{G} = (\tilde{V}, \tilde{E})] := \sum_{e \in \tilde{E}} W(e)$ is maximal and adheres to set of constraints M_c designed to

emulate multi-hop inference. The hypothesis h_i with the highest subgraph weight $W[\tilde{G} = (\tilde{V}, \tilde{E})]$ is selected to be the correct answer c_{ans}.

There are *two* major gaps with existing ILP-based QA solvers:

1. **Semantic Drift**: The challenge of a constraint-based explanation-based solver is to aggregate multiple facts. Each fact combined here is treated as a hop (i.e., *hopping* from one fact $\rightarrow$ fact or hypothesis $\rightarrow$ fact). With an increasing number of aggregated facts, the probability of inference drifting out of context also increases, leading to a phenomenon called *semantic drift*. Fried et al. (2015) and Jansen et al. (2018) had demonstrated that the performance gain achieved with 2-3 hops decreases for more than three hops.

2. **Non-differentiability of ILP-based solvers**: Constraint-based optimization solver provides a way to encode explicit and controllable assumptions to construct explanations and perform inference. While delivering explanations, existing optimization solvers are typically *non-differentiable* (Paulus et al., 2021) and cannot be integrated as part of a deep neural network. These approaches are also often limited by the number of constraints adopted for inference. This shortcoming prevents these methods from learning end-to-end on annotated corpora and achieving performance comparable to deep learning counterparts. Integrating constraint-based solvers with deep learning models can potentially combine the best of both worlds to achieve the following aims:

    - Acquire explanation-based inference, control and interpretability of constraint-based solvers into transformer-based models
    - Incorporate semantic flexibility supported by distributional semantics from transformer-based approaches into constraint-based solvers.

Given the above gaps, we formulate the **central problem** of the thesis as follows:

Problem Formulation:
Given a question (Q) and candidate answers $C = \{c_1, c_2, c_3, ..., c_n\}$, the aim is to build a differentiable constraint-based optimization model $Diff_{constrained}$ robust to semantic drift that combines constraint-based solvers and transformers to select the correct answer c_{ans} and explanation F_{ans} that supports the answer.

1.4 Research Objectives and Questions

The following section outlines the Research Objectives and Questions to address the central problem:

Systematic review of explanation-based multi-hop inference Explanation-based inference has emerged as a crucial requirement for multi-hop inference. However, little work has been done to present a systematic review of the field to identify the challenges and opportunities. Hence, the first primary research objective (**RO1**) we aim for is as follows:
RO1: *Identify challenges and opportunities within explanation-based multi-hop inference*

We seek to answer the following research questions by attempting to achieve Research Objective 1 (**RO1**):

- **RQ1.1**: *What types of inferences are required in multi-hop inference?*

  Aim to understand the different types of explanations presented in recent literature. Multi-hop inference is moving away from relying only on lexical overlaps and towards abstractive reasoning. Here, the solver is expected to go beyond the surface form of the problem and towards more abstract concepts. This RQ also aims to understand the effect of this paradigm in explanations.

- **RQ1.2**: *How have explanation-based benchmarks evolved to support multi-hop inference?*

  Investigate the benchmarks proposed for explanation-based multi-hop inference. Define a categorization for the benchmarks based on the domain, format, and explanation properties. Subsequently, group the explanation supporting benchmarks along the categories to analyze how the notion of inference represented in these datasets has evolved.

- **RQ1.3**: *How did explanation-based multi-hop inference models evolve?*

  Investigate different architectures used with explanation-based inference models. Identify different modelling paradigms based on the architectures used and plot how state-of-the-art has changed to arrive at the currently used modelling paradigm. Also, identify the challenges and shortcomings of each group of approaches.

- **RQ1.4**: *What are the gaps in the explanation-based multi-hop inference benchmarks and models?*

  Identify challenges and opportunities based on the analysis of the benchmarks and models. These findings allow us to outline the potential gaps that could be addressed with our research.

Tackling semantic drift in constraint-based solvers The current method of long hops of constrained solvers is not capable of dealing with semantic drift (Jansen et al., 2018; Khashabi et al., 2019).

Hence, the second primary research objective (**RO2**) we aim for is as follows:
RO2: *Propose a novel constraint-based solver for explanation-based multi-hop inference method which reduces semantic drift*

We seek to answer the following research question by attempting to achieve Research Objective 2 (**RO2**).

- **RQ2.1**: *Does the encoding of grounding-abstract mechanisms reduce semantic drift?*

  Empirically investigate if reducing the number of hops by encoding grounding-abstract mechanisms leads to better answer and explanation selection performance. Understand how each component plays a role in the inference process.

Integrating constraint-based solvers with transformer-based learning architectures to build Explanation-based multi-hop models Constraint-based solvers using Integer Linear Programming are non-differentiable and cannot be integrated into deep learning networks. Therefore, incorporating them in the current formulation is not possible and would require approximation (Agrawal, Amos, et al., 2019) or adaptation (Pogančić et al., 2019). We define the third research objective (*RO3*) as follows:
RO3: *Build a hybrid framework for multi-hop inference that combines constraint-based optimization layers with pre-trained neural representations, enabling end-to-end differentiability for explanation-based inference with optimization-based solvers.*

We use the same experiments with RO2 and seek to answer the following research question by attempting to achieve Research Objective 3 (**RO3**).

- **RQ3.1**: *Do incorporating constraint solvers with transformers improve performance when compared to the non-differentiable solver?*

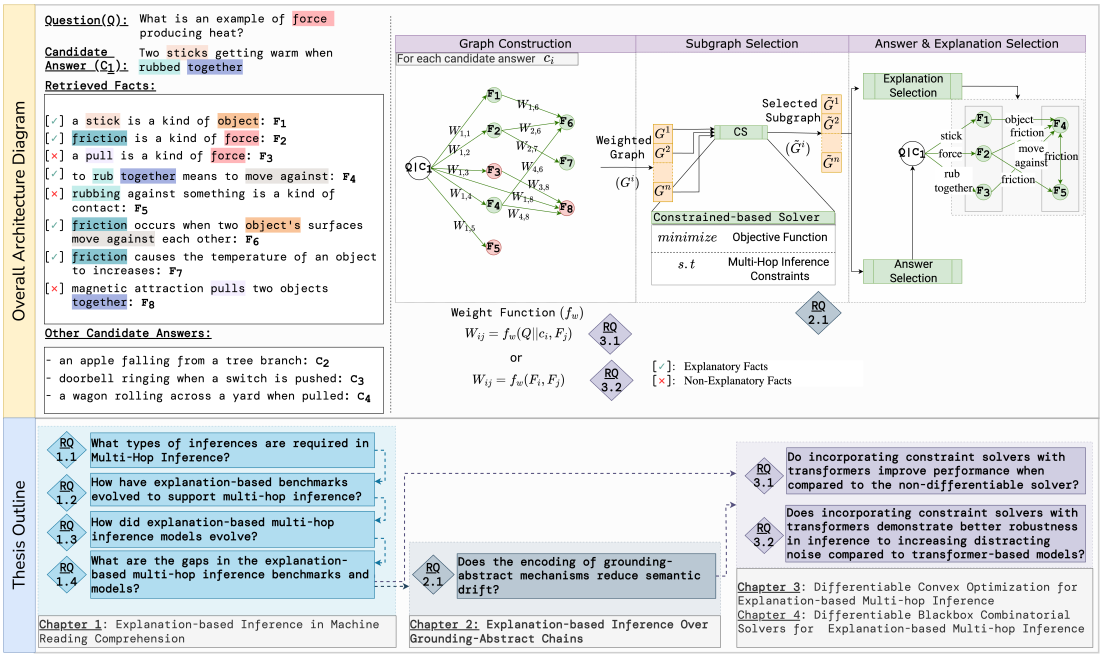

Figure 1.3: Overall end-to-end architecture diagram and how it connects to the research questions, the structure of the thesis, and dependencies between the chapters.

Compare the explanation-based inference performance obtained by a hybrid framework against the non-differentiable solver with an equivalent objective function and constraints.

- **RQ3.2**: *Does incorporating constraint solvers with transformers demonstrate better robustness in inference to increasing distracting noise compared to transformer-based models?*

  As noted by previous works (Yadav et al., 2019b, 2020), transformer-only models exhibit lower performance with increasing distraction information. With this RQ, we aim to compare the performance of transformer-based only models against our models with increasing distractors.

1.5 Thesis Outline

- **Chapter 2 (Explanation-based Inference in Machine Reading Comprehension)**:

  This Chapter presents the survey of recent benchmarks and approaches proposed for explanation-based machine reading comprehension (MRC). With MRC, we

cover both single-hop and multi-hop inference approaches. As dictated in Research Objective 1, this survey aims to present a systematic review of the field.

This Chapter is based on the paper "A Survey on Explainability in Machine Reading Comprehension". An earlier version of the paper can be found in https://arxiv.org/abs/2010.00389.

- **Chapter 3 (Explanation-based Inference Over Grounding-Abstract Chains)**:

  This Chapter proposes an explanation-based inference approach for science questions by reasoning on grounding and abstract inference chains. Our method, *ExplanationLP*, elicits explanations by constructing a weighted graph of relevant facts for each candidate answer and employs a linear programming formalism designed to select the optimal subgraph of explanatory facts. The graphs' weighting function comprises a set of parameters targeting relevance, cohesion and diversity, which we fine-tune for answer selection via Bayesian Optimization.

  This Chapter is based on the paper "Explainable Inference Over Grounding-Abstract Chains for Science Questions". The current version can be found in https://aclanthology.org/2021.findings-acl.1/ and has been accepted and published in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*.

- **Chapter 4 (Differentiable Convex Optimization for Explanation-based Multi-hop Inference)**:

  This Chapter presents *Diff*-Explainer, the first *hybrid* framework for an explanation-based multi-hop inference that integrates explicit constraints with neural architectures through differentiable convex optimization. Specifically, *Diff*-Explainer allows for fine-tuning neural representations within a constrained optimization framework to answer and explain multi-hop questions in natural language. To demonstrate the efficacy of the hybrid framework, we combine existing ILP-based solvers for multi-hop Question Answering (QA) with Transformer-based representations.

  This Chapter is based on the paper "Diff-Explainer: Differentiable Convex Optimization for Explainable Multi-hop Inference". This can be found in https://arxiv.org/pdf/2105.03417.pdf and has been accepted for *Transactions of the Association for Computational Linguistics, 2022*.

- **Chapter 5 (Differentiable Blackbox Combinatorial Solvers for Explanation-based Multi-hop Inference)**:

In Chapter 4, we proposed a novel methodology *Diff*-Explainer to integrate ILP with Transformers to achieve end-to-end differentiability for complex multi-hop inference. While this hybrid framework has been demonstrated to deliver better answer and explanation selection than transformer-based and existing ILP solvers, the neuro-symbolic integration still relies on a convex relaxation of the ILP formulation, which can produce sub-optimal solutions. To improve these limitations, we propose *Diff*-Comb Explainer, a novel neuro-symbolic architecture based on *Differentiable BlackBox Combinatorial solvers* (DBCS) (Pogančić et al., 2019). Unlike existing differentiable solvers, the presented model does not require the transformation and relaxation of the explicit semantic constraints, allowing for direct and more efficient integration of ILP formulations.

This Chapter is based on the paper "Going Beyond Approximation: Encoding Constraints for Explainable Multi-hop Inference via Differentiable Combinatorial Solvers". The current version of the paper can be found in https://arxiv.org/abs/2208.03339.

Figure 1.3 illustrates the overall end-to-end architecture diagram and how it connects to the research questions, the structure of the thesis, and the dependencies between the chapters.

1.6 Other Publications

- Mokanarangan Thayaparan, Marco Valentino, Peter Jansen, and Dmitry Ustalov. 2021. TextGraphs 2021 Shared Task on Multi-Hop Inference for Explanation Regeneration. In Proceedings of the Fifteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-15), pages 156–165, Mexico City, Mexico. Association for Computational Linguistics.

- Mokanarangan Thayaparan, Marco Valentino, Viktor Schlegel, and André Freitas. 2019. Identifying Supporting Facts for Multi-hop Question Answering with Document Graph Networks. In Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13), pages 42–51, Hong Kong. Association for Computational Linguistics.

- Guy Marshall, Mokanarangan Thayaparan, Philip Osborne, and André Freitas. 2021. Switching Contexts: Transportability Measures for NLP. In Proceedings of

the 14th International Conference on Computational Semantics (IWCS), pages 1–10, Groningen, The Netherlands (online). Association for Computational Linguistics.

- Marco Valentino, Mokanarangan Thayaparan, and André Freitas. 2021. Unification-based Reconstruction of Multi-hop Explanations for Science Questions. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pages 200–211, Online. Association for Computational Linguistics.

- Marco Valentino, Mokanarangan Thayaparan, and André Freitas. 2021. Hybrid Autoregressive Inference for Scalable Multi-hop Explanation Regeneration. Accepted to AAAI 2021

- Deborah Ferreira, Mokanarangan Thayaparan, Julia Rozanova, Marco Valentino, and André Freitas. 2021. To be or not to be an Integer?. Accepted to Findings of ACL 2022

- Deborah Ferreira, Julia Rozanova, Mokanarangan Thayaparan, Marco Valentino, and André Freitas. 2021. Does My Representation Capture X? Probe-Ably. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations, pages 194–201, Online. Association for Computational Linguistics.

- Julia Rozanova, Deborah Ferreira, Mokanarangan Thayaparan, Marco Valentino, and André Freitas. 2021. Supporting Context Monotonicity Abstractions in Neural NLI Models. In Proceedings of the 1st and 2nd Workshops on Natural Logic Meets Machine Learning (NALOMA), pages 41–50, Groningen, the Netherlands (online). Association for Computational Linguistics.

Chapter 2

Explanation-based Inference in Machine Reading Comprehension

This Chapter is based on the paper "A Survey on Explainability in Machine Reading Comprehension". An earlier version of the paper can be found in https://arxiv.org/abs/2010.00389. This Chapter is the literature review of the thesis.

2.1 Introduction

Machine Reading Comprehension (MRC) has the long-standing goal of developing machines that can reason with natural language. A typical reading comprehension task consists in answering questions about the background knowledge expressed in a textual corpus. Recent years have seen an explosion of models and architectures due to the release of large-scale benchmarks, ranging from open-domain (Rajpurkar et al., 2016; Z. Yang et al., 2018) to commonsense (Huang et al., 2019; Talmor et al., 2019) and scientific (Clark et al., 2018; Khot et al., 2020) reading comprehension tasks. Research in MRC is gradually evolving in the direction of abstractive inference capabilities, testing the models for their ability to go beyond what is explicitly stated in the text (Baral et al., 2020). As the need to evaluate abstractive reasoning becomes predominant, a crucial requirement emerging in recent years is *explanation-based inference* (Jansen et al., 2018; Khot et al., 2020; Xie et al., 2020; Z. Yang et al., 2018), intended as the ability of a model to expose the underlying mechanisms adopted to arrive at the final answers. Explanation-based inference has the potential to tackle some of the current issues in the field:

	Extractive MRC	**Abstractive MRC**
Question	When was Erik Watt's father born?	What is an example of a force producing heat?
Answer	May 5, 1939	Two sticks getting warm when rubbed together
Explanation	(1) He (Erik Watt) is the son of WWE Hall of Famer Bill Watts; (2) William F. Watts Jr. (born May 5, 1939) is an American former professional wrestler, promoter, and WWE Hall of Fame Inductee (2009).	(1) A stick is a kind of object; (2) To rub together means to move against; (3) Friction is a kind of force; (4) Friction occurs when two object's surfaces move against each other; (5) Friction causes the temperature of an object to increase.

Table 2.1: Explanations for extractive (Z. Yang et al., 2018) and abstractive (Jansen et al., 2018) MRC.

- **Evaluation:** Traditionally, MRC models have been evaluated on end-to-end answer prediction tasks. In other words, achieving high performance on specific datasets has been considered a proxy for evaluating the desired set of reasoning skills. However, recent work has demonstrated that this is not necessarily true for state-of-the-art models, which are particularly capable of exploiting biases in the data (Gururangan et al., 2018; McCoy et al., 2019). Research in explanation-based inference can provide novel evaluation frameworks to investigate and analyze the internal reasoning mechanisms (Dua et al., 2020; Inoue et al., 2020; Ross et al., 2017).

- **Interpretability:** A system capable of delivering explanations is generally more interpretable, meeting some of the requirements for real-world applications, such as user trust, confidence and acceptance (Biran & Cotton, 2017; Holzinger et al., 2017; Miller, 2019).

Despite the potential impact of explanation-based inference in MRC, little has been done to provide a unifying and organized view of the field. This Chapter aims to categorize explanation-supporting benchmarks and models systematically. To this end, we review the work published in some of the major AI and NLP conferences from 2015 onwards, which actively contributed to explanation-based inference in MRC, also referring to preprint versions when necessary.

2.2 Dimensions of Explanation-based Inference

As AI embraces a variety of tasks, the resulting definition of explanation-based inference is often fragmented and dependent on the specific scenario. Here, we frame the scope of the survey by investigating the dimensions of explanation-based inference in MRC.

We refer to *explanation-based inference* as a specialization of the higher level concept of *interpretability*. In general, interpretability aims at developing tools to understand and investigate the behaviour of an AI system. This definition also includes tools that are external to a black-box model, as in the case of post-hoc interpretability (Guidotti et al., 2018). On the other hand, the goal of explanation-based inference is the design of models capable of performing transparent inference through the generation of an *explanation* for the final prediction.

In general, an explanation can be seen as an answer to a *how* question formulated as follows: *"How did the model arrive at the conclusion c starting from the problem formulation p?"*. In the context of MRC, the answer to this question can be addressed by exposing the internal reasoning mechanisms linking p to c. This goal can be achieved in two different ways:

1. **Knowledge-based explanation:** exposing part of the relevant background knowledge connecting p and c in terms of supporting evidence and/or inference rules.

2. **Operational explanation:** composing a set of atomic operations through the generation of a symbolic program, whose execution leads to the final answer c.

This survey reviews recent developments in *knowledge-based* and *operational explanation*, emphasising the problem of *explanatory relevance* for the former – i.e., the identification of relevant information for the construction of explanations and *question decomposition* for the latter – i.e., casting a problem expressed in natural language into an executable program.

2.2.1 Explanation and Abstraction

Depending on the nature of the MRC problem, a complete explanation can include pieces of evidence at different levels of abstraction. Traditionally, the field has been divided into *extractive* and *abstractive* tasks (e.g. Table 2.1).

In extractive MRC, the reasoning required for the explanations is derivable from the original problem formulation. In other words, the correct decomposition of the problem provides the necessary inference steps for the answer, and the role of the explanation is

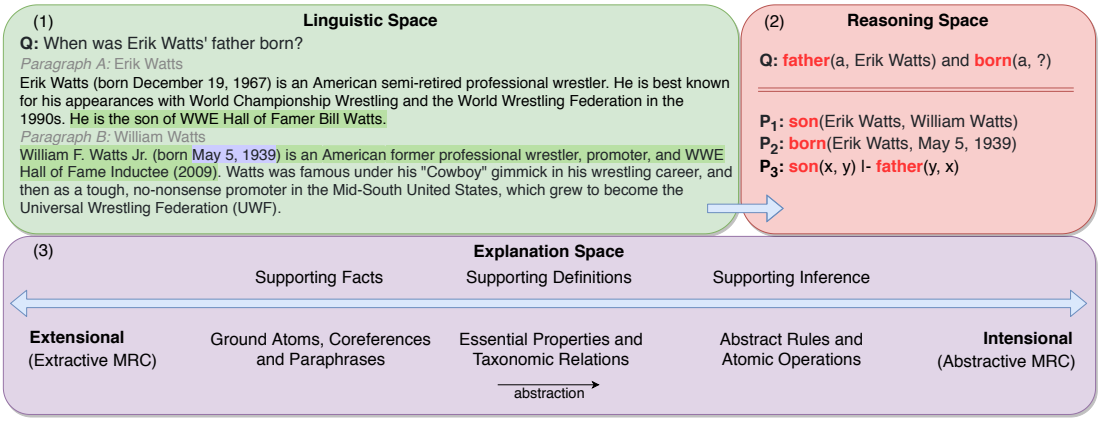

Figure 2.1: Dimensions of explanation-based inference in Machine Reading Comprehension.

to fill an information gap, identifying the correct arguments for a set of predicates via paraphrasing and co-reference resolution. As a result, explanations for extractive MRC are often expressed in the form of supporting passages retrieved from the contextual paragraphs (*extensional level*) (Z. Yang et al., 2018).

On the other hand, abstractive MRC tasks require going beyond the surface form of the problem with the inclusion of implicit knowledge about abstract concepts. In this case, the explanation typically leverages the use of supporting definitions, including taxonomic relations and essential properties, to perform abstraction from the original context in search of high-level rules and inference patterns (*intensional level*) (Jansen et al., 2016). As the nature of the task impacts explanation-based inference (See Fig. 2.1(3)), we consider the distinction between extractive and abstractive MRC throughout the survey, categorizing benchmarks and approaches according to the underlying reasoning involved in the explanations.

Domain	The knowledge domain – i.e. open domain (OD) , science (SCI), or commonsense (CS).
Format	The task format – i.e. span retrieval (Span), free-form (Free), multiple-choice (MC), textual entailment (TE).
MRC Type	The explanation can be derived from the correct decomposition of the problem – i.e. Extractive (Extr.); the explanation requires knowledge not expressed in the text – i.e. Abstractive (Abstr.).
Multi-hop (MH)	Whether the construction of explanations requires multi-hop reasoning – i.e. the aggregation of multiple pieces of evidence from the background knowledge.
Explanation Type (ET)	The type of explanation – i.e. knowledge-based (KB) or operational (OP).
Explanation Level (EL)	The explanations include only supporting facts – i.e. Extensional (E); the explanations expose the underlying inference rules or atomic operations – i.e. Intensional (I).
Background Knowledge (BKG)	The format of the provided background knowledge, if present, from which to extract or construct the explanations – i.e. single paragraph (SP), multiple paragraphs (MP), sentence corpus (C), table-store (TS), suit of atomic operations (AO).
Explanation Representation (ER)	The explanation representation – i.e. single passage (S), multiple passages (M), facts composition (FC), explanation graph (EG), generated sentence (GS), symbolic program (PR).

Dataset	Domain	Format	Type	MH	ET	EL	BKG	ER	Year
WikiQA (Y. Yang et al., 2015)	OD	Span	Extr.	N	KB	E	SP	S	2015
HotpotQA (Z. Yang et al., 2018)	OD	Span	Extr.	Y	KB	E	MP	M	2018
MultiRC (Khashabi, Chaturvedi, et al., 2018)	OD	MC	Abstr.	Y	KB	E	SP	M	2018
OpenBookQA (Mihaylov et al., 2018)	SCI	MC	Abstr.	Y	KB	I	C	FC	2018
Worldtree (Jansen et al., 2018)	SCI	MC	Abstr.	Y	KB	I	TS	EG	2018
e-SNLI (Camburu et al., 2018)	CS	TE	Abstr.	N	KB	I	-	GS	2018
Cos-E (Rajani et al., 2019)	CS	MC	Abstr.	N	KB	I	-	GS	2019
WIQA (Tandon et al., 2019)	SCI	MC	Abstr.	Y	KB	I	SP	EG	2019
CosmosQA (Huang et al., 2019)	CS	MC	Abstr.	N	KB	I	SP	S	2019
CoQA (Reddy et al., 2019)	OD	Free	Extr.	N	KB	E	SP	S	2019
Sen-Making (C. Wang et al., 2019)	CS	MC	Abstr.	N	KB	I	-	S	2019
ArtDataset (Bhagavatula et al., 2020)	CS	MC	Abstr.	N	KB	I	C	S,GS	2019
QASC (Khot et al., 2020)	SCI	MC	Abstr.	Y	KB	I	C	FC	2020
Worldtree V2 (Xie et al., 2020)	SCI	MC	Abstr.	Y	KB	I	TS	EG	2020
R^{4}C (Inoue et al., 2020)	OD	Span	Extr.	Y	KB	E	MP	EG	2020
Break (Wolfson et al., 2020)	OD	Free, Span	Abstr.	Y	OP	I	AO	PR	2020
R^3 (R. Wang et al., 2020)	OD	Free	Abstr.	Y	OP	I	AO	PR	2020

Table 2.2: Categorisation of *explanation-supporting* benchmarks in MRC.

2.3 Explanation-supporting Benchmarks

In this section, we review the benchmarks that have been designed for the development and evaluation of explanation-based reading comprehension models. Here, we consider only the benchmarks that exhibit at least one of the following properties:

1. **Labelled explanations:** The benchmark includes gold explanations that can be adopted as an additional signal for the development of explanation-based MRC models.

2. **Design for explanation evaluation:** The benchmark supports the use of quantitative metrics for evaluating the explanation-based inference of MRC systems, or it is explicitly constructed to test explanation-related inference.

For a complete overview of the existing datasets in MRC, the reader is referred to the following surveys: (Baradaran et al., 2020; B. Qiu et al., 2019; X. Zhang et al., 2019; Z. Zhang et al., 2020). The resulting classification of the benchmarks and the considered dimensions are described in Table 2.2.

2.3.1 Towards Abstractive MRC

In line with the general research trend in MRC, the development of explanation-supporting benchmarks is evolving towards evaluating abstractive reasoning, testing the models on their ability to go beyond the surface form of the text.

Explanation in early open-domain QA is framed as a single *sentence selection* problem (Y. Yang et al., 2015), where the evidence supporting the final answer is entirely encoded in a contiguous supporting passage. Subsequent work has started the transition towards tasks requiring multi-hop reasoning. HotpotQA (Z. Yang et al., 2018) is one of the first datasets designed to provide explicit annotation for the selection of multiple supporting facts, allowing for the development and evaluation of multi-hop and explanation-based inference models. The nature of HotpotQA is still closer to extractive MRC, where the structure of the explanations can be derived from the explicit decomposition of the questions (Min, Zhong, Zettlemoyer, et al., 2019). On the other hand, MultiRC (Khashabi, Chaturvedi, et al., 2018) combines multi-hop inference with various forms of abstract reasoning such as commonsense, causal relations, spatio-temporal and mathematical operations.

The annotation of supporting facts has demonstrated benefits in interpretability, bias reduction, and generalization in downstream tasks (Dua et al., 2020; Inoue et al.,

2020; Reddy et al., 2019). However, the gold explanations in these benchmarks are still expressed at the extensional level (See Fig.2.1), leaving implicit a consistent part of the underlying mechanisms adopted to derive the answer (Schlegel et al., 2020). To enrich the gold explanations, recent work has focused on operational interpretability, introducing explicit annotation for the decomposition of multi-hop and discrete reasoning questions (Dua et al., 2019) into a sequence of atomic operations (R. Wang et al., 2020; Wolfson et al., 2020).

The transition towards abstractive tasks has been supported by the development of large-scale benchmarks in the scientific domain (Clark, 2015; Clark et al., 2018), identified as a rich framework for evaluating explanation-based inference at the intensional level (Jansen et al., 2016). Explanations in the scientific domain, in fact, naturally mention facts about underlying regularities that require abstraction from the original problem formulation (Boratko et al., 2018). The benchmarks in this domain provide gold explanations for standardised science questions (Jansen et al., 2018; Mihaylov et al., 2018; Xie et al., 2020) or related scientific tasks such as what-if questions on procedural text (Tandon et al., 2019) and multi-hop sentence composition (Khot et al., 2020).

Recently, a set of abstractive tasks have been proposed for the evaluation of commonsense explanations (C. Wang et al., 2019). Cos-E (Rajani et al., 2019) and e-SNLI (Camburu et al., 2018) augment existing datasets for textual entailment (Bowman et al., 2015) and commonsense QA (Talmor et al., 2019) with crowd-sourced explanations, framing explanation-based inference as a natural language generation problem. Explanation-supporting benchmarks have now extended beyond question answering to hate speech detection (Mathew et al., 2021) and fake news detection (Dai et al., 2020). Other commonsense tasks have been explicitly designed to test explanation-related inference, such as causal and abductive reasoning (Huang et al., 2019). Bhagavatula et al. (2020) propose the tasks of Abductive Natural Language Inference (αNLI) and Abductive Natural Language Generation (αNLG), where MRC models are required to select or generate the hypothesis that best explains a set of observations.

2.3.2 Multi-hop Reasoning and Explanation

The construction of explanations in MRC typically requires multi-hop reasoning – i.e. the ability to compose multiple pieces of evidence to support the answer. However, the structure of the inference can differ according to the nature of the task.

In extractive MRC (Welbl et al., 2018; Z. Yang et al., 2018), multi-hop reasoning

often consists of identifying bridge entities or extracting and comparing information encoded in different passages. The explanations usually take the shape of linear chains or paths connecting distinct supporting facts via co-occurring Named Entities (e.g. *Bill Watts*, Table 2.1 left).

On the other hand, multi-hop reasoning for abstractive MRC aims to identify underlying rules or explanatory regularities that are not evident in the original problem. Jansen et al. (2018) observe that explanations for science questions require the construction of sentence graphs, in which each fact plays a specific role in the identification of core explanatory statements: *grounding facts* and *lexical glues* have the function of connecting the specific concepts in the question with their abstract semantic categories (e.g. *a stick is a kind of object*), while *central facts* refer to high-level explanatory knowledge (e.g. *friction causes the temperature of an object to increase*, Table 2.1 right). Similarly, explanations for multiple-choice questions in OpenbookQA (Mihaylov et al., 2018) require the retrieval of abstract scientific sentences, whose relevance can only be estimated by performing multi-hop reasoning through external commonsense knowledge.

Recent work suggests that the number of required hops for the explanations is correlated with *semantic drift* – i.e. the tendency of composing spurious inference chains that lead to wrong conclusions (Fried et al., 2015; Khashabi et al., 2019). The development of explanation-supporting benchmarks represents an attempt to limit this phenomenon by providing additional signals to learn abstract compositional schemes, thanks to the explicit annotation of valid inference chains (Jhamtani & Clark, 2020; Khot et al., 2020) or the extraction of common explanatory patterns to support the construction of many-hops explanations (Xie et al., 2020).

2.4 Explanation-based MRC Architectures

This section describes the major architectural trends for Explanation-based MRC (X-MRC). The approaches are broadly classified according to the MRC task they are applied to – i.e. extractive or abstractive. In order to elicit architectural trends, we further categorize the approaches as described in Table 2.3.

Figure 2.3 and 2.2 illustrate the resulting classification when considering the underlying architectural components for explanation generation. If an approach employs different modules for explanation generation and answer prediction, the latter is marked as $\triangle$. In some cases, an architecture can subsume a set of sub-components – e.g.

Explanation Type	(1) Knowledge-based explanation; (2) Operational-based explanation
Learning method	(1) Unsupervised (US): Does not require any annotated data; (2) Strongly Supervised (SS): Requires gold explanations for training or inference; (3) Distantly Supervised (DS): Treats explanation as a latent variable training only on problem-solution pairs.
Generated Output	Denotes whether the explanation is generated or composed from facts retrieved from the background knowledge.
Multi-Hop	Denotes whether the approach is designed for multi-hop reasoning

Table 2.3: Categories adopted for the classification of Explanation-based MRC approaches.

Transformers also includes attention networks. In cases like these, we only consider the larger component that subsumes the smaller one. If approaches employ both architectures, but as different functional modules, we plot them separately.

We generally observe an overall shift towards supervised methods over the years for both abstractive and extractive MRC. We posit that the advent of explanation-supporting datasets has facilitated the adoption of complex supervised neural architectures. Moreover, as shown in the classification, the majority of the approaches are designed for knowledge-based explanation. We attribute this phenomenon to the absence of large-scale datasets for operational interpretability until 2020. However, we note a recent uptake of distantly supervised approaches. We believe that further progress can be made with the introduction of novel datasets supporting symbolic question decomposition such as Break (Wolfson et al., 2020) and R^3 (R. Wang et al., 2020) (See Sec. 2.2).

While there are significant architectural overlaps between abstractive and extractive X-MRC, we do observe some key distinctions. Namely, we found no approaches that provided operational explanations for Abstractive X-MRC, and no models generated explanations for Extractive X-MRC. We hypothesize that the lack of operational explanations can be ascribed to the non-existence of Explanation-based Abstractive Operational datasets.

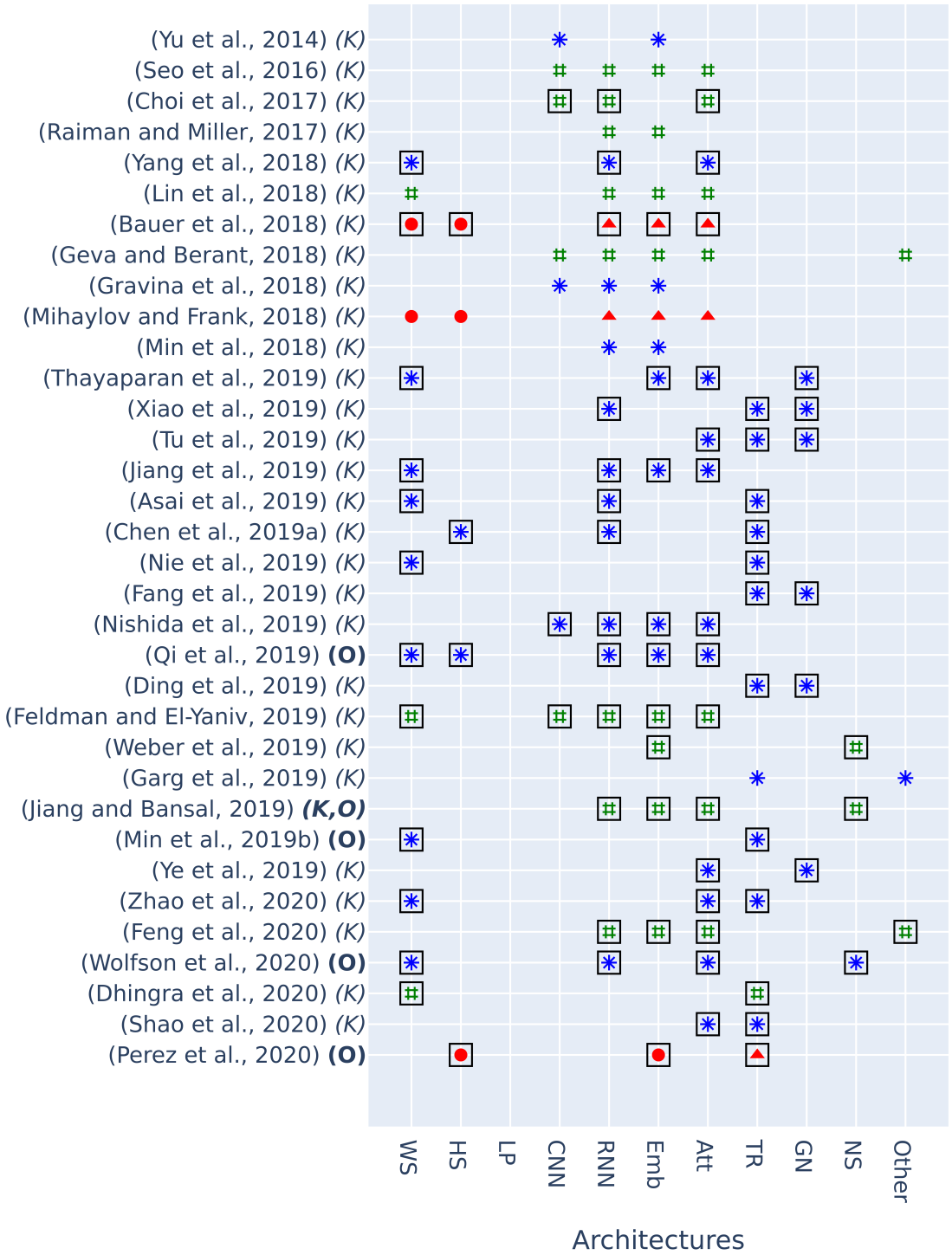

Figure 2.2: Explanation-based Extractive Machine Reading Comprehension (MRC) approaches. **Operational Explanations**: (**O**), **Knowledge-based Explanations**: (*K*), **Operational and Knowledge-based Explanations**: (***K,O***) **Learning**: Unsupervised (●), Distantly Supervised (#), Strongly Supervised (∗). **Generated Output**: (◯). **Multi Hop**: (▭). **Answer Selection Module:** (△). **Architectures**: WEIGHTING SCHEMES (WS): Document and query weighting schemes consist of information retrieval systems that use any form of vector space scoring system, HEURISTICS (HS): Hand-coded heuristics and scoring functions, INTEGER LINEAR PROGRAMMING (LP), CONVOLUTIONAL NEURAL NETWORK (CNN), RECURRENT NEURAL NETWORKS (RNN), PRE-TRAINED EMBEDDINGS (Emb), ATTENTION NETWORK (Att), TRANSFORMERS (TR), GRAPH NEURAL NETWORKS (GN), NEURO-SYMBOLIC (NS).

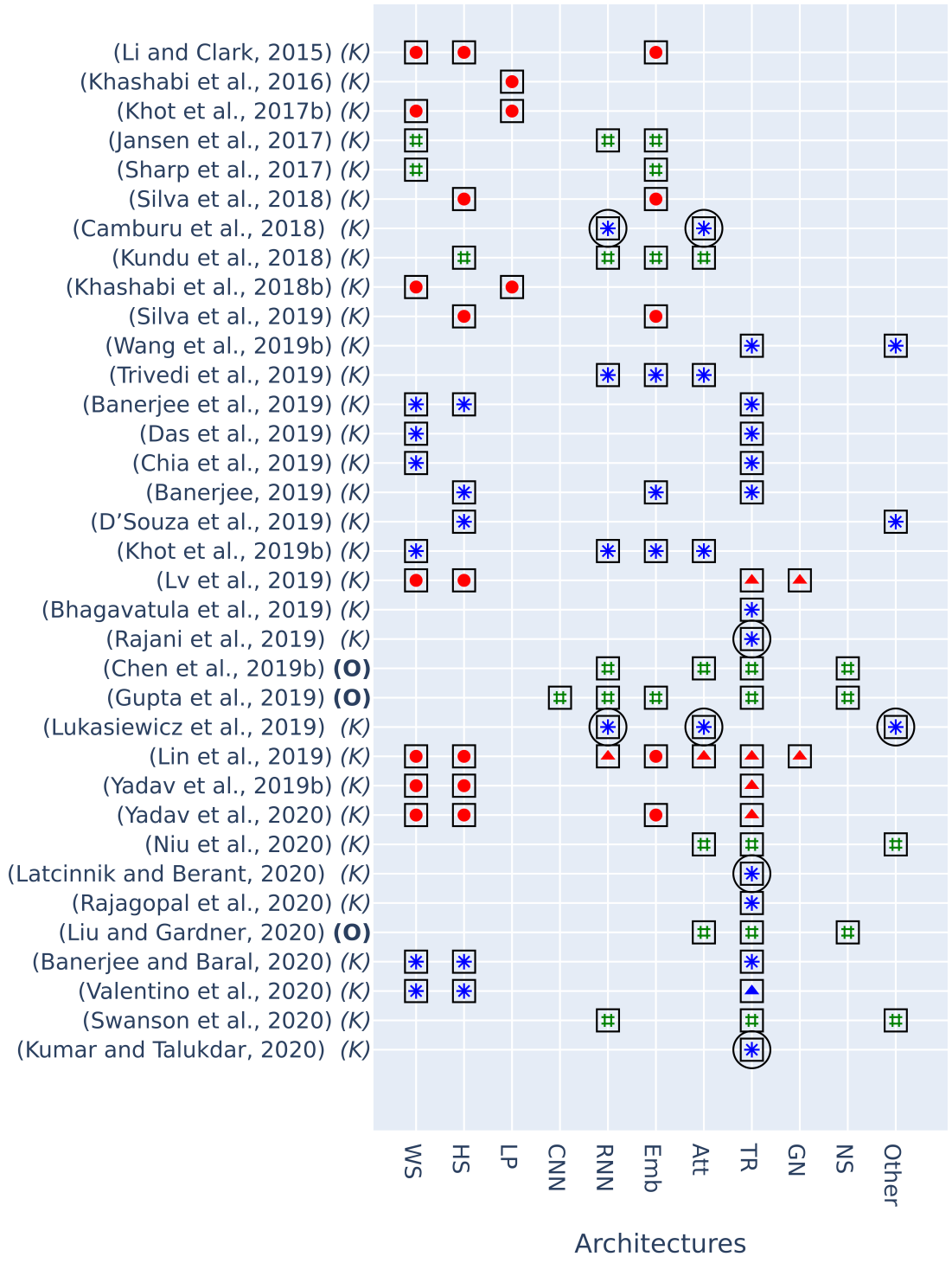

Figure 2.3: Explanation-based Abstractive Machine Reading Comprehension (MRC) approaches. **Operational Explanations**: (**O**), **Knowledge-based Explanations**: (*K*), **Operational and Knowledge-based Explanations**: (***K,O***) **Learning**: Unsupervised (●), Distantly Supervised (#), Strongly Supervised (∗). **Generated Output**: (◯). **Multi Hop**: (▭). **Answer Selection Module:** (△). **Architectures**: WEIGHTING SCHEMES (WS): Document and query weighting schemes consist of information retrieval systems that use any form of vector space scoring system, HEURISTICS (HS): Hand-coded heuristics and scoring functions, INTEGER LINEAR PROGRAMMING (LP), CONVOLUTIONAL NEURAL NETWORK (CNN), RECURRENT NEURAL NETWORKS (RNN), PRE-TRAINED EMBEDDINGS (Emb), ATTENTION NETWORK (Att), TRANSFORMERS (TR), GRAPH NEURAL NETWORKS (GN), NEURO-SYMBOLIC (NS).

2.4.1 Modeling Explanatory Relevance for Knowledge-based Explanations

Capturing relevance between the question-answer and fact, i.e., *explanatory relevance*, is imperative for knowledge-based explanations. Different approaches have devised distinct methods to encode explanatory relevance. This section reviews the approaches adopted for modelling explanatory relevance for *knowledge-based explanations*. We group the models into three main categories: *Explicit*, *Latent*, and *Hybrid*.

Explicit Models

Explicit models typically adopt heuristics and hand-crafted constraints to encode domain-specific hypotheses of explanatory relevance. The major architectural patterns are listed below:

Integer Linear Programming (ILP) Integer Linear programming has been used for modelling semantic and structural constraints in an unsupervised fashion. Early ILP systems, such as TableILP (Khashabi et al., 2016), formulate the construction of explanations as an optimal sub-graph selection problem over a set of semi-structured tables. Subsequent approaches (Khashabi, Khot, Sabharwal, & Roth, 2018; Khot et al., 2017) have proposed methods to reason over textual corpora via semantic abstraction, leveraging semi-structured representations automatically extracted through Semantic Role Labeling, Open Information Extraction, and Named Entity Recognition. Approaches based on ILP have been effectively applied for multiple-choice science questions when no gold explanation is available for strong supervision. While ILP based formulation has shown to provide control, they are *non-differentiable* and cannot be integrated as part of a broader deep learning architecture (Paulus et al., 2021; Pogančić et al., 2019).

Weighting schemes with heuristics The integration of heuristics and weighing schemes have been demonstrated to be effective for implementing lightweight methods that are inherently scalable to large corpora and knowledge bases. In the open-domain, approaches based on lemma overlaps and weighted triplet scoring function have been proposed (Mihaylov & Frank, 2018), along with path-based heuristics implemented with the auxiliary use of external knowledge bases (Bauer et al., 2018). Similarly, path-based heuristics have been adopted for commonsense tasks, where Lv et al. (2019)

propose a path extraction technique based on question coverage. For scientific and multi-hop MRC, Yadav et al. (2019b) propose ROCC, an unsupervised method to retrieve multi-hop explanations that maximize relevance and coverage while minimizing overlaps between intermediate hops.

Pre-trained embeddings with heuristics Pre-trained embeddings have the advantage of capturing semantic similarity, going beyond the lexical overlaps limitation imposed by the use of weighting schemes. This property has been shown to be useful for multi-hop and abstractive tasks, where approaches based on pre-trained word embeddings, such as GloVe (Pennington et al., 2014), have been adopted to perform semantic alignment between question, answer and justification sentences (Yadav et al., 2020). Silva et al. (2019), Silva et al. (2018) employ word embeddings and semantic similarity scores to perform selective reasoning on commonsense knowledge graphs and construct explanations for textual entailment.

Latent Models

Latent models learn the notion of explanatory relevance implicitly through machine learning techniques such as neural embeddings and language models. The architectural clusters adopting latent modelling are classified as follows:

Neural models for sentence selection This category refers to a set of neural approaches proposed for the *answer sentence selection* problem. These approaches typically adopt deep learning architectures, such as RNN, CNN and Attention networks via strong or distant supervision. Strongly supervised approaches (Garg et al., 2019; Gravina et al., 2018; Min et al., 2018; Yu et al., 2014) are trained on gold supporting sentences. In contrast, distantly supervised techniques (Raiman & Miller, 2017) indirectly learn to extract the supporting sentence by training on the final answer. Attention mechanisms have been frequently used for distant supervision (Seo et al., 2016) to highlight the attended explanation sentence in the contextual passage.

Transformers for multi-hop reasoning Transformers implement the encoder-decoder architecture. Given an input sequence $(x_1, \ldots, x_n)$ the encode module maps it a representation z ($z = (z_1, \ldots, z_n)$). The decoder then takes the representation z to generate output sequence y $z = (y_1, \ldots, y_n)$). The encoder and decoders are composed of multiple layers composed of multi-head attention mechanism and position-wise filly

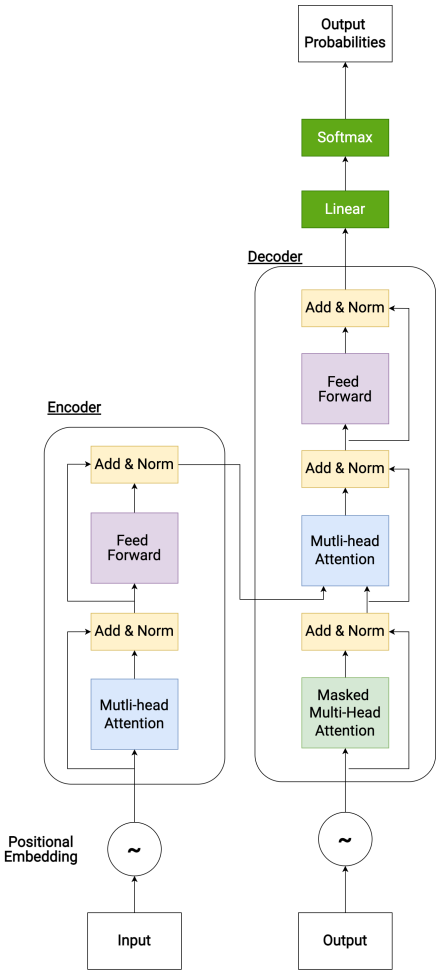

Figure 2.4: Encoder and Decoder of a Transformer model. Figure adapted from Vaswani et al. (2017)

connected feed-forward network. The architecture diagram of the transformer is illustrated in Figure 2.4.

Transformers are incapable of capturing sequence information. To alleviate this shortcoming, *positional encodings* injects a vector to the inputs. These vectors are based on specific periodic functions that the model uses to determine the position of the individual word.

Transformers uses Scaled Dot-Product Attention, that takes as input a set of queries Q and keys K dimension d_k, and values of dimension d_v:

$$Attention(Q,\ K,\ V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2.1}$$

Here, Q, K and V are obtained from the transformation over the input. Following

this step, multi-head attention expands this attention:

$$MultiHead(Q,K,V) = Concat(head_1,\ \ldots,\ head_h)W^O \tag{2.2}$$

where $head_i = Attention(QW_i^Q,\ KW_i^K,\ VW_i^V)$ and W_i^Q, W_i^K, W_i^V are projection parameter matrices.

The first model to leverage Transformer architectures to learn natural language representation in e Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019). BERT is pretrained on unlabeled data over two different tasks: Masked Langauge Modeling (MLM) and Next Sentence Prediction (NSP). The model predicts a masked word in a sentence for the MLM task. For the NSP task, the model is trained on a binary classification task of predicting if sentence B follows sentence A. This self-supervised learning is applied on large-scale corpora enabling the models to capture the syntactic and semantics of a language. BERT was trained on BooksCorpus (800M words) and English Wikipedia (2,500M words).

Different Transformer-based models have been proposed to improve on top of BERT. For example, RoBERTa (Y. Liu et al., 2019) removes the NSP training objective and adopts different hyperparameters achieving state-of-the-art performance over BERT. MPNet (Song et al., 2020) is another model that predicts tokens in random order instead of predicting tokens in sequential order enabling the capturing of bidirectional dependencies.

Transformers-based architectures have been successfully applied to learn explanatory relevance in both extractive and abstractive MRC tasks. Banerjee (Banerjee, 2019) and Chia et al. (2019) adopt a BERT model (Devlin et al., 2019) to learn to rank explanatory facts in the scientific domain. Shao et al. (2020) employ transformers with self-attention on multi-hop QA datasets (Z. Yang et al., 2018), demonstrating that the attention layers implicitly capture high-level relations in the text. The Quartet model (Rajagopal et al., 2020) has been adopted for reasoning on procedural text and producing structured explanations based on qualitative effects and interactions between concepts. In the distant supervision setting, Niu et al. (2020) address the problem of lack of gold explanations by training a self-supervised evidence extractor with auto-generated labels in an iterative process. Banerjee and Bara (Banerjee & Baral, 2020) propose a semantic ranking model based on BERT for QASC (Khot et al., 2020) and OpenBookQA (Mihaylov et al., 2018). Transformers have shown improved performance on downstream answer prediction tasks when applied in combination with explanations constructed through explicit models (Valentino et al., 2021; Yadav et al., 2019b, 2020).

Attention networks for multi-hop reasoning Similar to transformer-based approaches, attention networks have also been employed to extract relevant explanatory facts. However, attention networks are usually applied in combination with other neural modules. For HotpotQA, Z. Yang et al. (2018) propose a model trained in a multi-task setting on both gold explanations and answers, composed of recurrent neural networks and attention layers. Nishida et al. (2019) introduce a similarly structured model with a query-focused extractor designed to elicit explanations. The distantly supervised MUPPET model (Feldman & El-Yaniv, 2019) captures the relevance between question and supporting facts through bi-directional attention on sentence vectors encoded using pre-trained embedding, CNN, and RNN. In the scientific domain, Trivedi et al. (2019) re-purpose existing textual entailment datasets to learn the supporting facts relevance for multi-hop QA. Khot et al. (2019) propose a knowledge gap-guided framework to construct explanations for OpenBookQA.

Language generation models Recent developments in language modelling, along with the creation of explanation-supporting benchmarks, such as e-SNLI (Camburu et al., 2018) and Cos-E (Rajani et al., 2019), have opened up the possibility to generate semantically plausible and coherent explanation sentences automatically. Language models, such as GPT-2 (Radford et al., 2019), have been adopted for producing commonsense explanations, whose application has demonstrated benefits in terms of accuracy and zero-shot generalization (Latcinnik & Berant, 2020; Rajani et al., 2019). e-SNLI (Camburu et al., 2018) present a baseline based on a Bi-LSTM encoder-decoder with attention. Lukasiewicz et al. (2019) enhance this baseline by proposing an adversarial framework to generate more consistent and plausible explanations.

Hybrid Models

Hybrid models adopt heuristics and hand-crafted constraints as a pre-processing step to impose an explicit inductive bias for explanatory relevance. The major architectural patterns are listed below:

Graph Networks The relational inductive bias encoded in Graph Networks (Battaglia et al., 2018) provides viable support for reasoning and learning over structured representations. This characteristic has been identified as particularly suitable for supporting facts selection in multi-hop MRC tasks. A set of graph-based architectures have been proposed for multi-hop reasoning in HotpotQA (Z. Yang et al., 2018). Ye et al. (2019)

build a graph using sentence vectors as nodes and edges connecting sentences that share the same named entities. Similarly, Tu et al. (2019) construct a graph connecting sentences that are part of the same document, share noun phrases, and have named entities or noun phrases in common with the question. To improve scalability, the Dynamically Fused Graph Network (DFGN) (L. Qiu et al., 2019) adopts a dynamic graph construction, starting from the entities in the question and gradually selecting the supporting facts. Similarly, Ding et al. (2019) implement a dynamic graph exploration inspired by the dual-process theory (Evans, 1984, 2003; Sloman, 1996). The Hierarchical Graph Network (Fang et al., 2020) leverages a hierarchical graph representation of the background knowledge (i.e. question, paragraphs, sentences, and entities). In parallel with extractive MRC tasks, Graph Networks are applied for answer selection on commonsense reasoning, where a subset of approaches have started exploring the use of explanation graphs extracted from external knowledge bases through path-based heuristics (B. Y. Lin et al., 2019; Lv et al., 2019).

Explicit inference chains for multi-hop reasoning A subset of approaches have introduced end-to-end frameworks explicitly designed to emulate the step-by-step reasoning process involved in multi-hop MRC (J. Chen et al., 2019; Jiang et al., 2019; Kundu et al., 2019). The baseline approach proposed for Abductive Natural Language Inference (Bhagavatula et al., 2020) builds chains composed of hypotheses and observations and encodes them using transformers to identify the most plausible explanatory hypothesis. Similarly, Das et al. (2019) embeds the reasoning chains retrieved via TF-IDF and lexical overlaps using a BERT model to identify plausible explanatory patterns for multiple-choice science questions. In the open domain, Asai et al. (2019) build a graph structure using entities and hyperlinks and adopt recurrent neural networks to retrieve relevant documents sequentially. Nie et al. (2019) introduces a step-by-step reasoning process that retrieves the relevant paragraph, the supporting sentence, and the answer. Dhingra et al. (2020) propose an end-to-end differentiable model that uses Maximum Inner Product Search (MIPS) (Johnson et al., 2019) to query a virtual knowledge-base and extract a set of reasoning chains. Feng et al. (2020) propose a cooperative game approach to select the most relevant explanatory chains from a large set of candidates. In contrast to neural-based methods, Weber et al. (2019) propose a neuro-symbolic approach for multi-hop reasoning that extends the unification algorithm in Prolog with pre-trained sentence embeddings.

Neuro-Symbolic Reasoning A growing line of neuro-symbolic models focuses on adopting Transformers for interpretable reasoning over text (Clark et al., 2021; Gontier et al., 2020; Saha et al., 2020; Tafjord et al., 2021). For example, Saha et al. (2020) introduced the PROVER model that provides an interpretable transformer-based model that jointly answers binary questions over rules while generating the corresponding proofs. These models are related to the proposed framework for exploring hybrid architectures. Clark et al. (2021) proposed "soft theorem provers" operating over explicit theories in language. This hybrid reasoning solver integrates natural language rules with transformers to perform deductive reasoning. Saha et al. (2020) improved on top of it, enabling the answering of binary questions along with the proofs supporting the prediction. The multiProver (Saha et al., 2021) evolves on top of these conceptions to produce an approach that is capable of producing multiple proofs supporting the answer. While these hybrid reasoning approaches produce explainable and controllable inference, they assume the existence of natural language rules and have only been applied to synthetic datasets.

2.4.2 Operational Explanation

Operational explanations provide interpretability by exposing the set of operations adopted to arrive at the final answer. This section reviews the main architectural patterns for operational interpretability that focus on the problem of casting a question into an executable program.

Neural and Symbolic Programs Liu and Gardner (J. Liu & Gardner, 2020) propose a multi-step inference model with three primary operations: Select, Chain, and Predict. The Select operation retrieves the relevant knowledge; the Chain operation composes the background knowledge together; the Predict operation selects the final answer. Jiang and Bansel. (Jiang & Bansal, 2019b) propose the adoption of Neural Module Networks (Andreas et al., 2016b) for multi-hop QA by designing four atomic neural modules (Find, Relocate, Compare, NoOp) that allow for both operational explanation and supporting facts selection. Similarly, N. Gupta et al. (2020) adopt Neural Module Networks to perform discrete reasoning on DROP (Dua et al., 2019). In contrast, X. Chen et al. (2019) propose an architecture based on LSTM, attention modules, and transformers to generate compositional programs. While most of the neuro-symbolic approaches are distantly supervised, the recent introduction of question decomposition datasets (Wolfson et al., 2020) allows for direct supervision of symbolic program

generation (Subramanian et al., 2020).

Multi-hop question decomposition The approaches in this category aim at breaking multi-hop questions into single-hop queries that are simpler to solve. The decomposition allows for the application of divide-et-impera methods where the solutions for the single-hop queries are computed individually and subsequently merged to derive the final answer. Perez et al. (2020) propose an unsupervised decomposition method for the HotpotQA dataset. Min, Zhong, Zettlemoyer, et al. (2019) frame question decomposition as a span prediction problem adopting supervised learning with a small set of annotated data. Qi et al. (2019) propose GOLDEN Retriever, a scalable method to generate search queries for multi-hop QA, enabling the application of off-the-shelf information retrieval systems to select supporting facts.

2.5 Evaluation

The development of explanation-supporting benchmarks has allowed for a quantitative evaluation of the explanation-based inference in MRC. In open-domain settings, Exact Matching (EM) and F1 scores are often employed for evaluating the supporting facts (Z. Yang et al., 2018), while explanations for multiple-choice science questions have been evaluated using ranking-based metrics such as Mean Average Precision (MAP) (Jansen & Ustalov, 2019; Xie et al., 2020). In contexts where language models produce the explanations, natural language generation metrics have been adopted, such as BLEU score and perplexity (Papineni et al., 2002; Rajani et al., 2019).

Evaluating explanation-based inference through multi-hop reasoning still presents several challenges (J. Chen & Durrett, 2019; H. Wang, Yu, et al., 2019). Recent works have demonstrated that some of the questions in multi-hop QA datasets do not require multi-hop reasoning or can be answered by exploiting statistical shortcuts in the data (J. Chen & Durrett, 2019; Jiang & Bansal, 2019a; Min, Wallace, et al., 2019). In parallel, other works have shown that a consistent part of the expected reasoning capabilities for a proper evaluation of reading comprehension is missing in several benchmarks (Kaushik & Lipton, 2018; Schlegel et al., 2020). A set of possible solutions have been proposed, including the creation of evaluation frameworks for the gold standards (Schlegel et al., 2020), the development of novel metrics for multi-hop reasoning (Trivedi et al., 2020), and the adoption of adversarial training techniques (Jiang & Bansal, 2019a). Schuff et al. (2020) show that current models and evaluation metrics, such as the F1 score,

do not correlate with human experience, limiting the ability of the user to leverage the explanations for assessing the correctness of the system. The authors propose techniques to reinforce answer-explanation coupling with novel evaluation metrics better correlated with human judgment.

Regarding the evaluation of multi-hop program generation, Subramanian et al. (2020) observe that some of the modules in compositional neural networks (Andreas et al., 2016b), particularly suited for operational explanation-based inference, do not perform their intended behaviour, posing the problem of evaluating the faithfulness of the generated explanations. This problem can be alleviated by combining novel architectural design choices and auxiliary supervision.

2.6 Challenges and Opportunities

Benchmarks design and evaluation. Research on explanation-supporting benchmarks is progressing towards the design of abstractive tasks, with the inclusion of intensional elements in the gold explanations, such as generalized inference patterns for knowledge-based explanation inference (Jhamtani & Clark, 2020), and symbolic programs for operational-based explanation (Wolfson et al., 2020). However, it is essential to overcome the challenges regarding the evaluation of explanation-based inference (See Sec. 2.5). A gold standard evaluation should accompany the release of novel benchmarks, improving some of the emerging frameworks for the assessment of the reasoning capabilities involved in the MRC task (Kaushik & Lipton, 2018; Schlegel et al., 2020), together with the definition of evaluation metrics for the explanation-based inference that go beyond F1-score, and correlates with human judgment (Schuff et al., 2020; Trivedi et al., 2020). To this end, it is necessary to reinforce the connection with the field of Human-computer Interaction for the formalization and evaluation of different types of explanations (Miller, 2019; Sales et al., 2020). Additionally, verifying that it requires multi-step inference when proposing a new multi-hop reasoning benchmark is crucial. This verification can be achieved by testing the performance of strong one-hop baselines on the dataset (Min, Wallace, et al., 2019).

Scaling up annotated corpora. Additional research should be invested in the evaluation of semantic drift (Khashabi et al., 2019). Most of the existing multi-hop datasets only require the integration of up to 2 supporting sentences or paragraphs (Khot et al., 2020; Z. Yang et al., 2018). Empirical work has shown that semantic drift emerges

with long inference chains involving more than 2 hops (Fried et al., 2015). Therefore, developing additional datasets supporting the generation of long explanations is crucial. However, building corpora of explanations is costly as the annotators require approximately 60 hours of training for explanation authoring, in addition to the annotation and review process ($\approx$15 mins per training example) (Jansen et al., 2018). A possible direction to alleviate this problem is adopting a top-down approach by first defining a set of explanation templates representing common inference patterns and then re-using the templates for the annotation of gold explanations (Xie et al., 2020). A similar process with domain experts can help extend the creation of explanation supporting datasets in real-world scenarios such as medical, legal and regulatory settings.

Impact on inference and generalization. Integrating extracted explanations with downstream neural models has demonstrated promising results in performance and generalization across different domains (Rajani et al., 2019; Valentino et al., 2020; Yadav et al., 2019b). However, it is still unclear what aspect of the explanations helps downstream models achieve better performance. A crucial research direction, therefore, is the semantic probing of how different representations, knowledge categories and levels of abstraction impact downstream models and which types of explanations are useful to maximize the performance across different MRC tasks (Mitra et al., 2019; Tenney et al., 2019).

Knowledge-based architectures for knowledge-based explanations As observed in Section 2.4.1, integrating external knowledge bases is critical for explanation-based inference. A promising direction is using Graph Networks (Fang et al., 2020; L. Qiu et al., 2019) and end-to-end differentiable architectures over knowledge bases (Dhingra et al., 2020). However, most of these approaches are still limited to extractive MRC. An opportunity for future work is to extend these approaches to common sense and scientific reasoning tasks using recently developed resources such as GenericKB (Bhakthavatsalam et al., 2020).

Semantic control. Current language models are still limited by the generation of single-sentence explanations, lacking the semantic control to produce long inference chains, which are particularly important for abstractive MRC. Therefore, an open research question is whether new models, such as GPT-3 (Brown et al., 2020), can produce plausible and coherent multi-sentence explanations. A potential direction

to support the generation of multi-hop explanations is the adoption of explanation prototypes retrieved from training examples (Guu et al., 2018) or the learning of disentangled representations via Variational Auto Encoders (Norouzi et al., 2020).

Machine learning and symbolic reasoning. A promising direction for addressing explanation-based inference is the integration of neural models with symbolic reasoners. Recent approaches, such as NLProlog (Weber et al., 2019), have demonstrated that this integration is possible for multi-hop reasoning in MRC. These approaches are still limited to a maximum of two-hop reasoning, exhibiting lower performance when compared to state-of-the-art neural approaches. Therefore, additional research is required to improve the robustness of neuro-symbolic models and extend their applicability to more complex reading comprehension tasks. In this context, a promising research direction is adopting explanation corpora to learn the representation of generalized inference rules and integrate them with existing symbolic frameworks (Jhamtani & Clark, 2020) including ILP-based solvers (Khashabi, Khot, Sabharwal, & Roth, 2018; Khot et al., 2017).

2.7 Conclusion

Research Objective 1: *Identify challenges and opportunities within explanation-based multi-hop inference*

This survey analyzed existing benchmarks and models for explanation-based inference in Machine Reading Comprehension. We identified the emerging research trends and architectural design for explanation-based systems, highlighting challenges and opportunities for future work.

- **RQ1.1**: What types of inferences are required in multi-hop inference?

  In Section 2.2, we categorised the explanation-based on its function in inference into knowledge-based and operational-based. In addition to this categorization, we also presented how abstraction has emerged as a requirement for multi-hop inference solvers.

- **RQ1.2**: How have explanation-based benchmarks evolved to support multi-hop inference?

  In Section 2.3, we introduced a taxonomy to qualify explanation supporting benchmarks according to the domain, format and properties. We observe that

the development of explanation-supporting benchmarks is evolving towards evaluating abstractive reasoning and testing the models' ability to go beyond the surface form of the text. Our survey also identified *semantic drift* and how the development of explanation-supporting benchmarks represents an attempt to limit this phenomenon by providing additional signals to learn abstract compositional schemes, thanks to the explicit annotation of valid inference chains (Jhamtani & Clark, 2020; Khot et al., 2020), or the extraction of common explanatory patterns to support the construction of many-hops explanations (Xie et al., 2020).

- **RQ1.3**: How did explanation-based multi-hop inference models evolve?

  In Section 2.4, we selected a list of explanations providing multi-hop models and how the architectures used have changed. We also broadly classified these approaches into Explicit, Latent and Hybrid. We generally note a shift towards supervised methods over the years for both abstractive and extractive MRC. We note that hybrid models yield better performance on the knowledge-based explanation inference approaches. The current SOTA space of explanation-based inference is mainly composed of transformer-based models aided by hand-crafted constraints as a pre-processing step.

- **RQ1.4**: What are the gaps in the explanation-based multi-hop inference benchmarks and models?

  Finally, in Section 2.6, we presented some challenges and opportunities. We identified the effectiveness of hybrid approaches that combines latent representation with structural representation. We also identified the lack of semantic control and the need for neuro-symbolic models. In Section 2.4, we identified that constraint-based solvers based on ILP can provide semantic control and symbolic reasoning. However, little work has been done to integrate it with latent models. We also identified that these models have not explicitly addressed the perennial problem of semantic drift. These shortcomings formed the core motivation for the approaches and experiments in this thesis.

2.8 Scope and Limitations

Our survey was limited to the papers from ACL, EACL, NAACL, EMNLP, AAAI and Neurips. We also limited our survey to papers from the past five years. We imposed this scope to limit the number of research papers and enable the ease of filtering out a

high volume of highly cited research papers under one venue. We also omitted from analyzing the impact of explanations on generalisability as this was not the focus of the thesis.

Chapter 3

Explanation-based Inference Over Grounding-Abstract Chains

This Chapter is based on the paper "Explainable Inference Over Grounding-Abstract Chains for Science Questions". The current version can be found in https://aclanthology.org/2021.findings-acl.1/ and has been accepted and published in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*.

3.1 Introduction

Answering science questions remains a fundamental challenge in Natural Language Processing and AI as it requires complex forms of inference, including causal, model-based and example-based reasoning (Clark et al., 2018; Clark et al., 2013; Jansen, 2018; Jansen et al., 2016). Current state-of-the-art (SOTA) approaches for answering questions in the science domain are dominated by transformer-based models (Devlin et al., 2019; Sun et al., 2019). Despite remarkable performance on answer prediction, these approaches are black-box by nature, lacking the capability of providing *explanations* for their predictions (Biran & Cotton, 2017; Jansen et al., 2016; Miller, 2019).

Explanation-based Science Question Answering solvers typically treat explanation generation as a multi-hop graph traversal problem. Here, the solver attempts to compose multiple facts that connect the question to a candidate answer. These *multi-hop* approaches have shown diminishing returns with an increasing number of hops (Jansen, 2018; Jansen et al., 2018). Fried et al. (2015) conclude that this phenomenon is due to *semantic drift* – i.e., as the number of aggregated facts increases, so does the probability

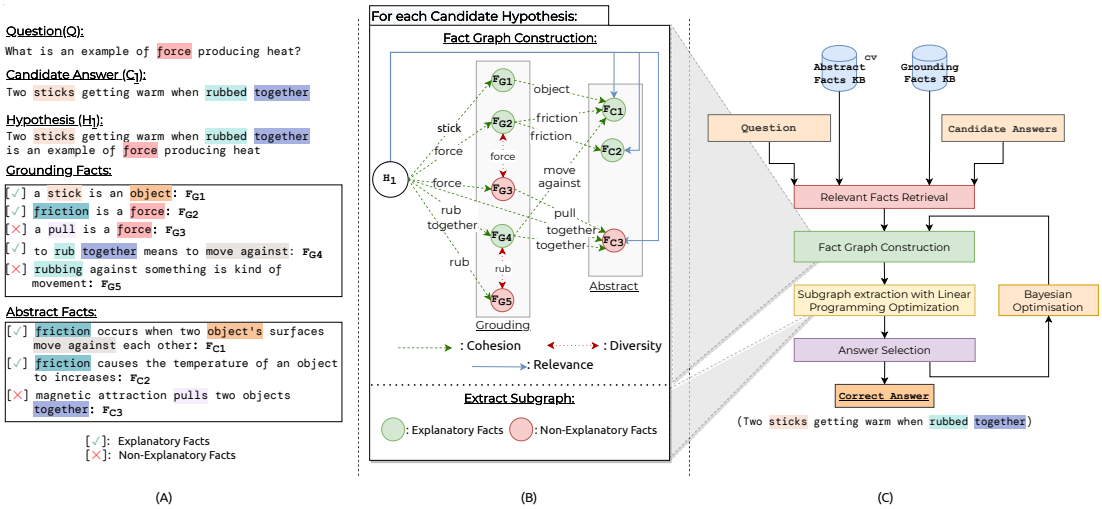

Figure 3.1: Overview of our approach: (A) Depicts a question, answer and formulated hypothesis along with the set of facts retrieved from a fact retrieval approach (B) Illustrates the optimization process behind extracting explanatory facts for the provided hypothesis and facts. (C) Details the end-to-end architecture diagram.

of inference drifting out of context. Khashabi et al. (2019) propose a theoretical framework, empirically supported by Fried et al. (2015) and Jansen et al. (2018), attesting that ongoing efforts with *very long* multi-hop reasoning chains are unlikely to succeed, emphasizing the need for a *richer* representation with fewer hops and higher importance to abstraction and grounding mechanisms.

Consider the example in Figure 3.1A where the central concept the question examines is the understanding of *friction*. Here, an inference solver's challenge is to identify the core scientific facts (**Abstract Facts**) that best explain the answer. To achieve this goal, a QA solver should be able first to go from *force* to *friction*, *stick* to *object* and *rubbing together* to *move against*. These are the **Grounding Facts** that link generic or abstract concepts in a core scientific statement to specific terms occurring in question and candidate answer (Jansen et al., 2018). The grounding process is followed by the identification of the abstract facts about *friction*. A complete explanation for this question would require the composition of five facts to derive the correct answer successfully. However, it is possible to reduce the global reasoning in two hops, modelling it with grounding and abstract facts.

In line with these observations, this work presents a novel approach that explicitly models abstract and grounding mechanisms. The contributions of the Chapter are:

1. We present a novel approach that performs explanation-based reasoning via

grounding-abstract chains combining Integer Linear Programming with Bayesian optimization for science question answering (Section 3.2).

2. We obtain comparable performance when compared to transformers, multi-hop approaches and previous Integer Linear Programming models despite having a significantly lower number of parameters (Section 3.3.1).
3. We demonstrate that our model can generate plausible explanations for answer prediction (Section 3.3.2) and validate the importance of grounding-abstract chains via ablation analysis (Section 3.3.4).

3.2 ExplanationLP: Explanation-based Inference with Integer Linear Programming

ExplanationLP answers and explains multiple-choice science questions via explanation-based inference. Specifically, the task of answering multiple-choice science questions is reformulated as the problem of finding the candidate answer that is supported by the best explanation. For each Question Q and candidate answer $c_i \in C$, ExplanationLP converts to a hypothesis h_i and attempts to construct a plausible explanation.

Figure 3.1C illustrates the end-to-end framework. From an initial set of facts selected using a retrieval model, ExplanationLP constructs a fact graph where each node is a fact, and the nodes and edges have a score according to three properties: *relevance*, *cohesion* and *diversity*. Subsequently, an optimal subgraph is extracted using ILP, whose role is to select the best subset of facts while preserving structural constraints imposed via grounding-abstract chains. The subgraphs' global scores computed by summing up the nodes and edges scores are adopted to select the final answer. Since the subgraph scores depend on the sum of nodes and edge scores, each property is multiplied by a learnable weight optimized via Bayesian optimization to obtain the best possible combination with the highest accuracy for answer selection. To the best of our knowledge, we are the first to combine a parameter optimization method with ILP for inference. The rest of this section describes the model in detail.

3.2.1 Relevant facts retrival

Given a question (Q) and candidate answers $C = \{c_1,\ c_2,\ c_3, ...,\ c_n\}$ we convert them to hypotheses $\{h_1,\ h_2,\ h_3, ...,\ h_n\}$ using the approach proposed by Demszky et al. (2018).

For each hypothesis h_i we adopt fact retrieval approaches (e.g: BM25, Unification-retrieval (Valentino et al., 2021)) to select the top m relevant *abstract* facts $F_A^{h_i} = \{f_1^{h_i}, f_2^{h_i}, f_3^{h_i}, ..., f_m^{h_i}\}$ from a knowledge base containing abstract facts (*Abstract Facts KB*) and top l relevant *grounding* facts $F_G^{h_i} = \{f_1^{h_i}, f_2^{h_i}, f_3^{h_i}, ..., f_l^{h_i}\}$ from a knowledge base containing grounding facts (*Grounding Facts KB*) that *at least* connects one abstract fact with the hypothesis, such that $F^{h_i} = F_A^{h_i} \cup F_G^{h_i}$ and $l + m = k$.

3.2.2 Fact graph construction

For each hypothesis h_i we build a weighted undirected graph $G^{h_i} = (V^{h_i}, E^{h_i}, \omega_v, \omega_e)$ with vertices $V^{h_i} \in \{\{h_i\} \cup F^{h_i}\}$, edges E^{h_i}, edge-weight function $\omega_e(e_i; \theta_1)$ and node-weight function $\omega_v(v_i; \theta_2)$ where $e_i \in E^{h_i}$, $v_i \in V^{h_i}$ and $\theta_1, \theta_2 \in [0, 1]$ is a learnable parameter which is optimized via Bayesian optimization.

The model scores the nodes and edges based on the following *three* properties (See Figure 3.1B):

1. **Relevance**: We promote the inclusion of highly relevant facts in the explanations by encouraging the selection of sentences with higher lexical relevance and semantic similarity with the hypothesis. We use the following scores to measure the relevance and the semantic similarity of the facts:

   Lexical Relevance score (*L*): Obtained from the upstream facts retrieval model (e.g: BM25 score/ Unification score (Valentino et al., 2021)).

   Semantic Similarity score (*S*): Cosine similarity obtained from neural sentence representation models. For our experiments, we adopt Sentence-BERT (Reimers et al., 2019) since it shows state-of-the-art performance in semantic textual similarity tasks.

2. **Cohesion**: Explanations should be cohesive, implying that grounding-abstract chains should remain within the same context. To achieve cohesion, we encourage a high degree of overlaps between different hops (e.g. hypothesis-grounding, grounding-abstract, hypothesis-abstract) to prevent the inference chains from drifting away from the original context. The overlap across two hops is quantified using the following scoring function:

   Cohesion score (*C*): We denote the set of unique terms of a given fact $f_i^{h_i}$ as $t(f_i^{h_i})$ after being lemmatized and stripped of stopwords. The overlap score of

two facts $f_j^{h_i}$ and $f_j^{h_i}$ is given by:

$$C(f_j^{h_i}, f_k^{h_i}) = \frac{|t(f_j^{h_i}) \cap t(f_k^{h_i})|}{max(|t(f_j^{h_i})|, |t(f_k^{h_i})|)} \tag{3.1}$$

Therefore, the higher the number of term overlaps, the higher the cohesion score.

3. **Diversity**: While maximizing relevance and cohesion between different hops, we encourage diversity between facts of the same type (e.g. abstract-abstract, grounding-grounding) to address different parts of the hypothesis and promote completeness in the explanations. We measure diversity via the following function:

   Diversity score (*D*): We denote the overlaps between hypothesis h_i and the fact $f_i^{h_i}$ as $t_{h_i}(f_i^{h_i}) = t(f_i^{h_i}) \cap t(h_i)$. The diversity score of two facts $f_j^{h_i}$ and $f_j^{h_i}$ is given by:

$$D(f_j^{h_i}, f_k^{h_i}) = -1 \frac{|t_{h_i}(f_j^{h_i}) \cap t_{h_i}(f_k^{h_i})|}{max(|t_{h_i}(f_j^{h_i})|, |t_{h_i}(f_k^{h_i})|)} \tag{3.2}$$

   The goal is to maximize diversity and avoid redundant facts in the explanations. Therefore, if two facts overlap with different parts of the hypothesis, they will have a higher diversity score compared to two facts that overlap with the same part.

Given these premises, the weight functions of the graph is designed as follows:

$$\omega_e(v_j, v_k; \theta_1) = \begin{cases} \theta_{gg} D(v_j, v_k) & v_j, v_k \in F_G^{h_i} \\ \theta_{aa} D(v_j, v_k) & v_j, v_k \in F_A^{h_i} \\ \theta_{ga} C(v_j, v_k) & v_j \in F_G^{h_i}, v_k \in F_A^{h_i} \\ \theta_{qg} C(v_j, v_k) & v_j \in F_G^{h_i}, v_k = h_i \\ \theta_{qa} C(v_j, v_k) & v_j \in F_A^{h_i}, v_k = h_i \end{cases} \tag{3.3}$$

$$\omega_v(v_i^{h_i}; \theta_2) = \begin{cases} \theta_{lr} L(v_j, h_i) + \theta_{ss} S(v_j, h_i) & v_j \in F_A^{h_i} \\ 0 & v_i \in F_G^{h_i} \\ 0 & v_i = h_i \end{cases} \tag{3.4}$$

where θ_{gg}, θ_{aa}, θ_{ga}, θ_{gq}, $\theta_{qa} \in \theta_1$ and θ_{lr}, $\theta_{ss} \in \theta_2$.

3.2.3 Subgraph extraction with Integer Linear Programming (ILP) optimization

The construction of the explanation graph has to be optimized for the downstream answer selection task. Specifically, from the whole set of facts retrieved by the upstream retrieval models, we need to select the optimal subgraph that maximizes the performance of answer prediction. To achieve this goal, we adopt an ILP approach.

The selection of the explanation graph is framed as a rooted maximum-weight connected subgraph problem with a maximum number of K vertices (R-MWCS$_K$). This formalism is derived from the generalized maximum-weight connected subgraph problem (Loboda et al., 2016). R-MWCS$_K$ has two parts: objective function to be maximized and constraints to build a connected subgraph of explanatory facts. The formal definition of the objective function is as follows:

Definition 1. Given a connected undirected graph $G = (V, E)$ with edge-weight function $\omega_e : E \to \mathbb{R}$, node-weight function $\omega_v : V \to \mathbb{R}$, root vertex $r \in V$ and expected number of vertices K, the rooted maximum-weight connected subgraph problem with K number of vertices (R-MWCS$_K$) problem is finding the connected subgraph $\hat{G} = (\hat{V}, \hat{E})$ such that $r \in \hat{V}$, $|V| \leq K$ and

$$\Omega(\hat{G}; \theta_3) = \theta_{vw} \sum_{v \in \hat{V}} \omega_v(v; \theta_1) + \theta_{ew} \sum_{e \in \hat{E}} \omega_e(e; \theta_2) \to max \tag{3.5}$$

where θ_{vw}, $\theta_{ew} \in \theta_3$, $\theta_3 \in [0, 1]$ and θ_3 is a learnable parameter optimized via Bayesian optimization. The LP solver will seek to extract the optimal subgraph with the highest possible sum of node and edge weights. Since the solver seeks to obtain the highest possible score, it will avoid negative edges and prioritize high-value positive edges resulting in higher diversity, cohesion and relevance. We adopt the following binary variables to represent the presence of nodes and edges in the subgraph:

1. Binary variable y_v takes the value of 1 iff $v \in V^{h_i}$ belongs to the subgraph.
2. Binary variable z_e takes the value of 1 iff $e \in E^{h_i}$ belongs to the subgraph.

In order to emulate the grounding-abstract inference chains and obtain a valid subgraph, we impose the set constraints described as follows:

Chaining constraint: Equation 3.6 states that the subgraph should always contain the question node. Inequality 3.7 states that if a vertex is to be part of the subgraph, then at

least one of its neighbours with a lexical overlap should also be part of the subgraph. Equation 3.6 and Inequality 3.7 restrict the LP system to construct explanations that originate from the question and perform multi-hop aggregation based on the existence of lexical overlap. Inequalities 3.8, 3.9 and 3.10 state that if two vertices are in the subgraph then the edges connecting the vertices should be also in the subgraph. These inequality constraints will force the LP system to avoid grounding nodes with high overlap regardless of their relevance.

$$y_{v_i} = 1 \qquad \text{if } v_i = h_i \tag{3.6}$$

$$y_{v_i} \leq \sum_j y_{v_j} \qquad \forall v_j \in N_{G^{h_i}}(v_i) \tag{3.7}$$

$$z_{v_i,v_j} \leq y_{v_i} \qquad \forall e_{(v_i,v_j)} \in E \tag{3.8}$$

$$z_{v_i,v_j} \leq y_{v_j} \qquad \forall e_{(v_i,v_j)} \in E \tag{3.9}$$

$$z_{v_i,v_j} \geq y_{v_i} + y_{v_j} - 1 \qquad \forall e_{(v_i,v_j)} \in E \tag{3.10}$$

Abstract fact limit constraint: Equation 3.11 limits the total number of abstract facts to M. By limiting the abstract facts, we dictate the need for grounding facts based on the number of entities present in the question and the abstract facts.

$$\sum_i y_{v_i} \leq M \qquad \forall v_i \in F_A^{h_i} \tag{3.11}$$

Grounding neighbour constraint: Inequality 3.12 states that if a grounding fact is selected, then at least two of its neighbours should be either both abstract facts or a question and an abstract fact. This constraint ensures that grounding facts play the linking role in connecting question-abstract or abstract-abstract.

$$\sum_{v_j} y_{v_i} - 2 \geq -2(1 - y_{v_j}) \qquad \begin{array}{r} \forall v_i \in N_{G^{h_i}}(v_j), \\ v_i \in \{F_A^{h_i} \cup h_i\}, \\ v_j \in F_G^{h_i} \end{array} \tag{3.12}$$

3.2.4 Bayesian optimization for Answer Selection

Given Question Q and choices $C = \{c_1, c_2, c_3, ..., c_n\}$ we extract the optimal explanation graphs $\hat{G}^Q = \{\hat{G}^{c_1}, \hat{G}^{c_2}, \hat{G}^{c_3}, ..., \hat{G}^{c_n}\}$ for each choice. We consider the hypothesis with the highest relevance, cohesion and diversity to be the correct answer. Based on this premise we define the correct answer as $c_{ans} = \text{argmax}_{h_i} \ (\Omega(\hat{G}^{h_i}))$.

In order to automatically optimize the Integer Linear Programming model (i.e, θ_1, θ_2, θ_3) we use Bayesian optimization.

Bayesian optimization is a branch of machine-learning-based optimization. Bayesian optimization is applied to optimize objective functions that are expensive to evaluate. It builds a surrogate function with a Gaussian prior for the objective function. The uncertainty in the surrogate function is quantified using Bayesian machine learning techniques and Gaussian regression. It uses the acquisition function to decide which space to sample next to maximize the performance of the objective function (Frazier, 2018). Bayesian optimization has been applied for hyperparameter tuning for machine learning (Klein et al., 2017; Snoek et al., 2012), reinforcement learning (Brochu et al., 2010; Wilson et al., 2014) and algorithm configuration (Hutter et al., 2011).

Bayesian optimization (BayesOpt) is focused on solving the following problem:

$$\hat{x} = \underset{x \in \mathbb{X}}{\text{argmax}} \, f(x) \tag{3.13}$$

where the aim is to find $\hat{x}$ that maximizes the function $f(x)$ over some domain $\mathbb{X}$. Bayesian optimization attempts to find the maximum point with a minimum number of evaluations. $f(x)$ is usually "expensive-to-evaluate", and the number of iterations is usually limited to a few hundred.

BayesOpt is composed of two main components:

- **Probabilistic model of the function**: to model the objective function. The probabilistic model is widely represented by the Gaussian process. With each observation $(x_t, f(x_t))$, we learn the distribution and obtain the posterior.

- **Acquisition function**: to decide where to sample next in order maximise the function f. The function samples the next point based on its definition and might favour exploration or exploitation.

Overall, the Bayesian optimization algorithm can be defined as follows:

Algorithm 1: Bayesian optimization Algorithm

Define Gaussian Prior ($\mathcal{GP}$) to model f
Evaluate f at n_0 points
$n = n_0$
while $n \leq N$ **do**
  Update the posterior probability of $\mathcal{GP}$ based on the n data point evaluations
  Get the next exploration point x_n from the acquisition function, where the acquisition function is computed using the current posterior distribution.
  $y_n = f(x_n)$.
  $n = n + 1$
end
Return a solution: either the point $\hat{x}$ evaluated with the largest $f(x)$

Gaussian process regression Gaussian process is used for modelling functions in Bayesian statistics. Given a finite collection of points $x_1,\ x_2,\ x_3,\ \cdots,\ x_k \in \mathbb{R}^d$ we can collect the function scores along these points as $[f(x_1),\ f(x_2),\ \cdots,\ f(x_k)]$. Gaussian regression assumes that these points were drawn from a prior multivariate normal distribution with a particularly mean and covariance matrix.

The mean vector is calculated by executing the mean function μ_0 at each point x_i. On the other hand, the covariance matrix is constructed by a covariance function or kernel Σ_0 at each point $(x_i,\ x_j)$. The covariance function is chosen so that two closer points have a high positive correlation. Additionally, the covariance matrix is also expected to be positive semi-definite.

Given these assumptions, the prior distribution of $[f(x_1),\ f(x_2),\ \cdots,\ f(x_k)]$ is defined as,

$$f(x_{1:k}) \ \sim \ Normal(\mu_0(x_{1:k}),\ \Sigma_0(x_{1:k},x_{1:k})) \tag{3.14}$$

Based on this assumption, given that we have observed $f(x_{1:n})$ without noise and try to infer the value of the function at a new point x, the conditional distribution (i.e: *posterior probability distribution*) can be calculated as:

$$\begin{aligned} f(x) \mid f(x_{1:n}) &= Normal(\mu_n(x^2), \sigma^2(x)) \\ \mu_n(x) &= \Sigma_0(x,x_{1:n})\Sigma_0(x_{1:n},x_{1:n})^{-1}(f(x_{1:n}) - \mu_0(x_{1:n})) + \mu_0(x) \\ \sigma^2(x) &= \Sigma_0(x,x) - \Sigma_0(x,x_{1:n})\Sigma_0(x_{1:n},x_{1:n})^{-1}\Sigma_0(x_{1:n},x) \end{aligned} \tag{3.15}$$

Given three points $x, x^{'}, x^{''}$, with the property $\|x - x^{'}\| < \|x - x^{''}\|$, the kernel function

should have the property $\Sigma_0(x,x^{'}) > \Sigma_0(x,x^{''})$. Following are some widely used kernels:

- **RBF Kernel**: $\Sigma_0(x_i,x_j) = \exp\left(-\frac{d(x_i,x_j)^2}{2l^2}\right)$ where l is the length scale of the kernel and d is the Euclidean distance.
- **Mattern Kernel**: $\Sigma_0(x_i,x_j) = \frac{1}{\Gamma(\nu)2^{\nu-1}}\left(\frac{\sqrt{2\nu}}{l}d(x_i,x_j)\right)^{\nu} K_{\nu}\left(\frac{\sqrt{2\nu}}{l}d(x_i,x_j)\right)$ where where d is the Euclidean distance, K_{ν} is a modified Bessel function and Γ is the gamma function.
- **Rational Quadratic Kernel**: $\Sigma_0(x_i,x_j) = \left(1+\frac{d(x_i,x_j)^2}{2\alpha l^2}\right)^{-\alpha}$ where α is the scale mixture parameter, l is the length scale of the kernel and d is the Euclidean distance

Acquisition Functions The next part of the optimization is to define the acquisition function. The acquisition function decides the next exploration point. The acquisition function's challenge is finding the trade-off between exploration and exploitation.

Below are some of the popular acquisition functions:

- **Upper Confidence Bound (UCB)** (Srinivas et al., 2009)

$$UCB(x^{'}) = \mu(x^{'}) + \beta^{\frac{1}{2}}\sigma(x^{'}) \tag{3.16}$$

  UCB selects regions with larger μ for exploitation and large σ for exploration. The parameter β is used to balance between the two.

- **Probability of Improvement (PI)** (Kushner, 1964):

$$PI(x^{'}) = \int_{f(\hat{x})}^{\infty} Norm_{f(x^{'})}(\mu(x^{'}),\sigma(x^{'}))df(x^{'}) \tag{3.17}$$

  PI computes the likelihood at $x^{'}$ so that it will result in higher than the current maximum $f(\hat{x})$.

- **Expected Improvement (EI)** (Močkus, 1975):

$$EI(x^{'}) = \int_{f(\hat{x})}^{\infty} (f(x^{'}) - f(\hat{x}))Norm_{f(x^{'})}(\mu(x^{'}),\sigma(x^{'}))df(x^{'}) \tag{3.18}$$

Unlike PI, EI prefers larger improvements over smaller and highly likely improvements. The expectation improvement $f(x^{'}) - f(\hat{x})$ is calculated of the normal distribution that is above the current maximum.

For our approach, we empirically found that RationalQuadratic Kernel for Gaussian Process and Expected Improvement acquisition function yield the best results.

Given the details above, the algorithm to select the correct answer with Bayesian Optimization is defined below.

Algorithm 2: Bayesian optimization for Answer Selection.

$\theta_1, \theta_2, \theta_3$ = initRandom(seed)
$G^{\mathcal{Q}}$ = fact-graph-construction($\omega_e(\theta_1^{'}),\ \omega_v(\theta_2^{'})$)
$\hat{G}^{\mathcal{Q}}$ = LP($G^{\mathcal{Q}},\ \Omega(\theta_3)$)
X = evaluate-accuracy($G^{\mathcal{Q}}$)
model = $\mathcal{GP}(X, \{\theta_1, \theta_2, \theta_3\})$
iteration = 0
while *iteration* $\leq N$ **do**
  $\theta_1^{'}, \theta_2^{'}, \theta_3^{'}$ = get-next-exploration-point()
  $G^{\mathcal{Q}^{'}}$ = fact-graph-construction($\omega_e(\theta_1^{'}),\ \omega_v(\theta_2^{'})$)
  $\hat{G}^{\mathcal{Q}^{'}}$ = LP($G^{\mathcal{Q}'}$,$\Omega(\theta_3^{'})$)
  $X^{'}$ = evaluate-accuracy($G^{\mathcal{Q}^{'}}$)
  model.update($X^{'}, \{\theta_1^{'}, \theta_2^{'}, \theta_3^{'}\}$)
  iteration = iteration + 1
end
Result: Best accuracy for model and respective parameters $\theta_1, \theta_2, \theta_3$

3.3 Empirical Evaluation

Background Knowledge: We construct the required knowledge bases using the following sources.

1. **Abstract KB**: Our Abstract knowledge base is constructed from the WorldTree Tablestore corpus (Jansen et al., 2018; Xie et al., 2020). The Tablestore corpus contains a set of common sense and scientific facts adopted to create explanations for multiple-choice science questions. The corpus is built for answering elementary science questions encouraging possible knowledge reuse to elicit explanatory patterns. We extract the core scientific facts to build the Abstract KB.

Core scientific facts are independent from the specific questions and represent general scientific and commonsense knowledge, such as `Actions` (*friction occurs when two object's surfaces move against each other*) or `Affordances` (*friction causes the temperature of an object to increase*).

2. **Grounding KB**: The grounding knowledge base consists of definitional knowledge (e.g., synonymy and taxonomy) that can take into account lexical variability of questions and help it link it to abstract facts. To achieve this goal, we select the *is-a* and *synonymy* facts from ConceptNet (Speer et al., 2017) as our grounding facts. ConceptNet has high coverage and precision, enabling us to answer a wide variety of questions.

Question Sets: We use the following question sets to evaluate ExplanationLP's performance and compare it against other explainable approaches:

1. **WorldTree Corpus** (Jansen et al., 2018): The 2,290 questions in the WorldTree corpus are split into three different subsets: *train-set* (987), *dev-set* (226) and *test-set* (1,077). We use the *dev-set* to assess the explanation selection performance and robustness analysis since the explanations for *test-set* are not publicly available.

2. **ARC-Challenge Corpus**: ARC-Challenge is a multiple-choice question dataset consisting of questions from science exams from grade 3 to grade 9 (Clark et al., 2018). We only consider the Challenge set of questions, and these questions have proven to be challenging to answer for other LP-based question answering and neural approaches.

Relevant Facts Retrieval (FR): We experiment with two different fact retrieval scores. The first model – i.e. *BM25 Retrieval*, adopts a BM25 vector representation for hypothesis and explanation facts. We apply this retrieval for both Grounding and Abstract retrieval. We use the IDF score from BM25 as our downstream model's relevance score. The second approach – i.e. *Unification Retrieval (UR)*, represents the BM25 implementation of the Unification-based Reconstruction framework described in Valentino et al. (2021). The unification score for a given fact depends on how often the same fact appears in explanations for similar questions.

Baselines: The following baselines are replicated on the WorldTree corpus to compare against ExplanationLP:

1. **Bert-Based models**: We compare the ExplanationLP model's performance against a set of BERT baselines. The first baseline – i.e. $BERT_{Base}$/$BERT_{Large}$, is represented by a standard BERT language model (Devlin et al., 2019) fine-tuned for multiple-choice question answering. Specifically, the model is trained for binary classification on each question-candidate answer pair to maximize the correct choice (i.e., predict 1) and minimize the wrong choices (i.e., predict 0). During inference, we select the choice with the highest prediction score as the correct answer. BERT baselines are further enhanced with explanatory facts retrieved by the retrieval models. *BERT + BM25* and *BERT + UR*, is fine-tuned for binary classification by complementing the question-answer pair with grounding and abstract facts selected by BM25 and Unification retrieval, respectively.

2. **PathNet** (Kundu et al., 2019): PathNet is a neural approach that constructs a single linear path composed of two facts connected via entity pairs for reasoning. PathNet also can explain its reasoning via explicit reasoning paths. They have exhibited strong performance for multiple-choice science questions while adopting a two-hop reasoning strategy. Similar to BERT-based models, we employ PathNET with the top k facts retrieved utilizing Unification (*PathNet + UR*) and BM25 (*PathNet + BM25*) retrieval.

Further details regarding the hyperparameters and code used for each model, along with information concerning the knowledge base construction and dataset information, can be found in the Supplementary Materials.

3.3.1 Answer Selection

WorldTree Corpus: We retrieve the top l relevant grounding facts from Grounding KB and the top m relevant abstract facts from Abstract KB such that $l+m=k$ and $l=m$. To ensure fairness across the approaches, the same amount of facts are presented to each model. We experimented with $k=\{10,\ 20,\ 30,\ 40,\ 50\}$ and reported the accuracy across Easy and Challenge split of the best performing setting in Table 3.1. We draw the following conclusions:

1. Despite having a smaller number of parameters to train (BERT_{Base}: 110M parameters, BERT_{Large}: 340M parameters, ExplanationLP: 9 parameters), the best performing ExplanationLP (#10) overall outperforms all the BERT_{Base} and

#	Model	Accuracy	
		Easy	Challenge
1	BERT_{Base}	51.04	28.75
2	BERT_{Large}	54.58	29.39
3	BERT_{Base} + BM25 (k=10)	53.92	42.72
4	BERT_{Large} + BM25 (k=10)	54.05	43.45
5	BERT_{Base} + UR (k=10)	52.87	42.17
6	BERT_{Large} + UR (k=10)	58.50	43.72
7	PathNet + BM25 (k=20)	43.32	36.42
8	PathNet + UR (k=15)	47.64	33.55
9	Ours + BM25 (k=30)	63.82	48.24
10	Ours + UR (k=30)	**66.23**	**50.15**

Table 3.1: Accuracy on Easy (764) and Challenge split (313) of WorldTree *test-set* corpus from the best performing k of each model

BERT_{Large} models on both Challenge and Easy split. We outperform the best-performing BERT model with facts (BERT_{Large} (#6)) by 7.74% in Easy and 6.43% in Challenge. We also outperform best performing BERT without facts (BERT_{Large} (#2)) by 11.66% in Easy and 20.76% in Challenge.

2. BERT is inherently a black-box model, not being entirely possible to explain its prediction. By contrast, ExplanationLP is fully explainable and produces a complete explanatory graph.

3. Similar to ExplanationLP, PathNet is also explainable and demonstrates robustness to noise. ExplanationLP also outperforms PathNet's best performance setting (#8) by 18.59% in Easy and 16.60% in Challenge.

4. ExplanationLP consistently exhibits better scores on both BM25 and UR than BERT and PathNet, demonstrating independence of the upstream retrieval model for performance.

ARC-Challenge : We also evaluated our model on the ARC-Challenge corpus (Clark et al., 2018) to evaluate ExplanationLP on a more extensive general question set and compare against contemporary approaches that provide explanations for an inference that has *only* been trained on ARC corpus. Table 3.2 reports the results on the *test-set*. We compare ExplanationLP against published approaches that are fully/partly

#	Model	Explainable	Accuracy
1	BERT_{Large}	No	35.11
2	IR Solver (Clark et al., 2016)	Yes	20.26
3	TupleInf (Khot et al., 2017)	Yes	23.83
4	TableILP (Khashabi et al., 2016)	Yes	26.97
5	DGEM (Clark et al., 2016)	Partial	27.11
6	KG^2 (Y. Zhang et al., 2018)	Partial	31.70
7	ET-RR (Ni et al., 2019)	Partial	36.61
8	Unsupervised AHE (Yadav et al., 2019a)	Partial	33.87
9	Supervised AHE (Yadav et al., 2019a)	Partial	34.47
10	AutoRocc (Yadav et al., 2019b)	Partial	**41.24**
11	Ours + BM25 (k=40)	Yes	**40.21**
12	Ours + UR (k=40)	Yes	39.84

Table 3.2: ARC challenge scores compared with other Fully or Partially explainable approaches trained *only* on the ARC dataset.

CASE I: All the selected facts are in the gold explanation (**Frequency**: 33%)
Question: A company wants to make a game that uses a magnet that sticks to a board. Which material should it use for the board? **Answer:** steel **Explanations:** (1) steel is a metal (*Grounding*), (2) if a magnet is attracted to a metal then that magnet will stick to that metal (*Abstract*), (3) a magnet attracts magnetic metals through magnetism (*Abstract*),
CASE II: At least one selected facts are in the gold explanation (**Frequency**: 58%)
Question: A large piece of ice is placed on the sidewalk on a warm day. What will happen to the ice? **Answer**: It will melt to form liquid water. **Explanations:** (1) drop is liquid small amount (*Grounding*), (2) forming something is change (*Grounding*), (3) ice wedging is mechanical weathering (*Grounding*), (4) melting means changing from a solid into a liquid by adding heat energy (*Abstract*), (5) weathering means breaking down surface materials from larger whole into smaller pieces by weather (*Abstract*),
CASE III: No retrieved facts is in the gold explanation (**Frequency**: 9%)
Question: Wind is a natural resource that benefits the southeastern shore of the Chesapeake Bay. How could these winds best benefit humans? **Answer:** The winds could be converted to electrical energy **Explanations:** (1) renewable resource is natural resource (*Grounding*), (2) wind is a renewable resource (*Abstract*), (3) electrical devices convert electricity into other forms of energy (*Abstract*)

Table 3.3: Case study of explanation extracted by ExplanationLP

explainable. Here explainability indicates if the model produces an explanation/evidence for the predicted answer. A subset of the approaches produces evidence for the answer but remains intrinsically black-box. These models have been marked as *Partial*.

As depicted in the Table 3.2, we outperform the best performing fully explainable (#4 TableILP) model by 13.28%. We also outperform specific neural approaches with larger parameter sets (#5 - #9) that provide explanations for their inference and BERT (#1). Despite having a smaller number of training parameters, we also exhibit competitive performance with a state-of-the-art Bert-based approach (#10) that do not use external resources to train the QA system.

3.3.2 Explanation Selection

Table 3.4 shows the Precision, Recall and $F1_{Macro}$ score for explanation retrieval for PathNet and ExplanationLP. These scores are computed using gold abstract explanations

Approach	Precision	Recall	F1
PathNet + UR (k=20)	21.56	36.55	29.06
Ours + UR (k=30)	**57.96**	**49.92**	**48.13**

Table 3.4: Explanation retrieval performance on the WorldTree Corpus *dev-set*.

from WorldTree corpus. We outperform PathNet across all spectrums by a significant margin.

Table 3.3 reports three representative cases that show how explanation generation relates to correct answer prediction. The first example (Case I) represents the situation in which all the selected sentences are annotated as gold explanations in the WorldTree corpus (dev-set). The second example (Case II) shows the case in which at least one sentence in the explanation is labelled as gold. Finally, the third example (Case III) represents the case in which the explanation generated by the method does not contain any gold fact. We observe Case I and Case II occur over 91% of the questions, demonstrating that the correct answers are mostly derived from plausible explanations.

3.3.3 Robustness

Distracting knowledge Table 3.5 reports the analyzes carried out with BERT, PathNet and ExplanationLP on the WorldTree *test-set* for varying top k relevant facts to measure the robustness towards increasing distractors. These scores, along with an overall steady drop in performance with increasing k indicate that BERT struggles with the increasing number of distracting knowledge. In contrast, ExplanationLP can operate on a higher amount of distracting information and still obtain better scores, displaying resilience towards the noise. The lowest drop in performance of ExplanationLP with BM25 is 1.33% (#9 k=30 $\rightarrow$ k=10) and UR is 1.12% (#10 k=30 $\rightarrow$ k=10). On the other hand, the lowest drop in performance for BERT$_{Large}$ with BM25 is 24.32% (#5 k=10, $\rightarrow$ k=50) and UR is 27.95% (#6, k=10 $\rightarrow$ k=50). While PathNet also exhibits resilience to noise, our approach still outperforms it by a significant margin across all settings.

Hypothesis complexity Figure 3.2 presents the change in accuracy as the number of unique terms increases in hypothesis and answer. Our approach demonstrates lower degradation in performance when compared to PathNet. While the drop in BERT performance is lower than ours, we still outperform BERT across the entire spectrum. These observations show the robustness of our approach with regard to hypothesis complexity.

#	Model	Accuracy				
		k=10	k=20	k=30	k=40	k=50
3	BERT$_{Base}$ + BM25	50.92	43.63	37.97	31.56	32.68
4	BERT$_{Base}$ + UR	49.76	42.14	31.84	30.36	31.29
5	BERT$_{Large}$ + BM25	51.25	43.36	32.86	35.46	26.92
6	BERT$_{Large}$ + UR	54.50	42.98	27.39	24.88	26.55
7	PathNet + BM25	41.02	41.61	41.98	40.11	41.79
8	PathNet + UR	42.36	43.58	40.76	41.22	42.83
9	ExplanationLP + BM25	58.00	58.23	59.33	59.21	59.05
10	ExplanationLP + UR	**60.25**	**60.63**	**61.37**	**61.28**	**61.00**

Table 3.5: Overall answer selection performance on the WorldTree *test-set*. k represents the number of retrieved facts by the respective retrieval approaches.

Semantic Drift To validate the performance across an increasing number of hops, we plot the accuracy against explanation length as illustrated in Figure 3.3. As demonstrated in explanation regeneration (Jansen & Ustalov, 2019; Valentino et al., 2021), the complexity of a science question is directly correlated with the explanation length – i.e. the number of facts required in the gold explanation. Unlike BERT, PathNet and ExplanationLP use external background knowledge, addressing the multi-hop process in two main reasoning steps. However, in contrast to ExplanationLP, PathNet combines only two explanatory facts to answer a given question. This assumption has a negative impact on answering complex questions requiring long explanations. This is evident in the graph, where we observe a sharp decrease in accuracy with increasing explanation length. Comparatively, ExplanationLP achieves a more stable performance, showing a lower degradation with an increasing number of explanation sentences. These results crucially demonstrate the positive impact of grounding-abstract mechanisms on semantic drift. We also exhibit consistently better performance when compared with BERT as well.

3.3.4 Ablation Study

In order to understand the contribution lent by different components, we choose the best setting (*WorldTree*: ExplanationLP + UR (k=30) and *ARC*: ExplanationLP + BM25 (k=40)) and drop different components to perform an ablation analysis. We retain the ensemble after removing each component. The results are summarized in Table 3.6.

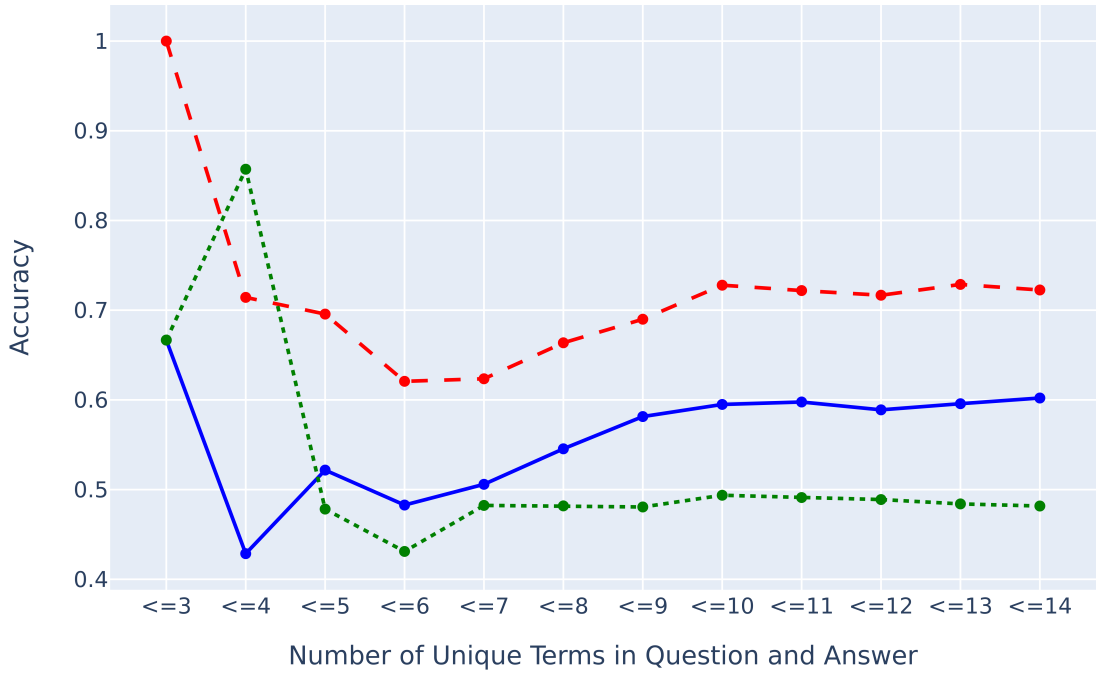

Figure 3.2: Change in accuracy of answer prediction the development set varying across different models with increasing unique terms in hypothesis for WorldTree *dev-set*. Red dashed line represents ExplanationLP + UR (k=30), blue line represents BERT$_{Large}$ + UR (k=10) and green dotted line represents PathNet + UR (k=20)

1. The grounding-abstract chains (#2) play a significant role, particularly in the reasoning mechanism on a challenging question set like ARC-Challenge.

2. As observed in #3, #4 removing node weights and edge weights lead to a dramatic drop in performance. This drop indicates that both are fundamental for the final prediction, highlighting the role of graph structure in explainable inference.

3. The importance of cohesion varies across different types of facts. We observe that Hypothesis-Abstract cohesion (#5) is significantly more important than the others. We attribute this to the fact that without Hypothesis-Abstract cohesion, multi-hop inference can quickly go out of context.

4. From the ablation analysis, we can see how lexical relevance and semantic similarity (#10, 11) complement each other towards the final prediction. For the WorldTree corpus, the relevance score has a higher parameter score translating into a higher impact and vice-versa for ARC.

5. Diversity plays a more minor role when compared to cohesion and relevance. The impact of diversity in ARC is higher than that of WorldTree.

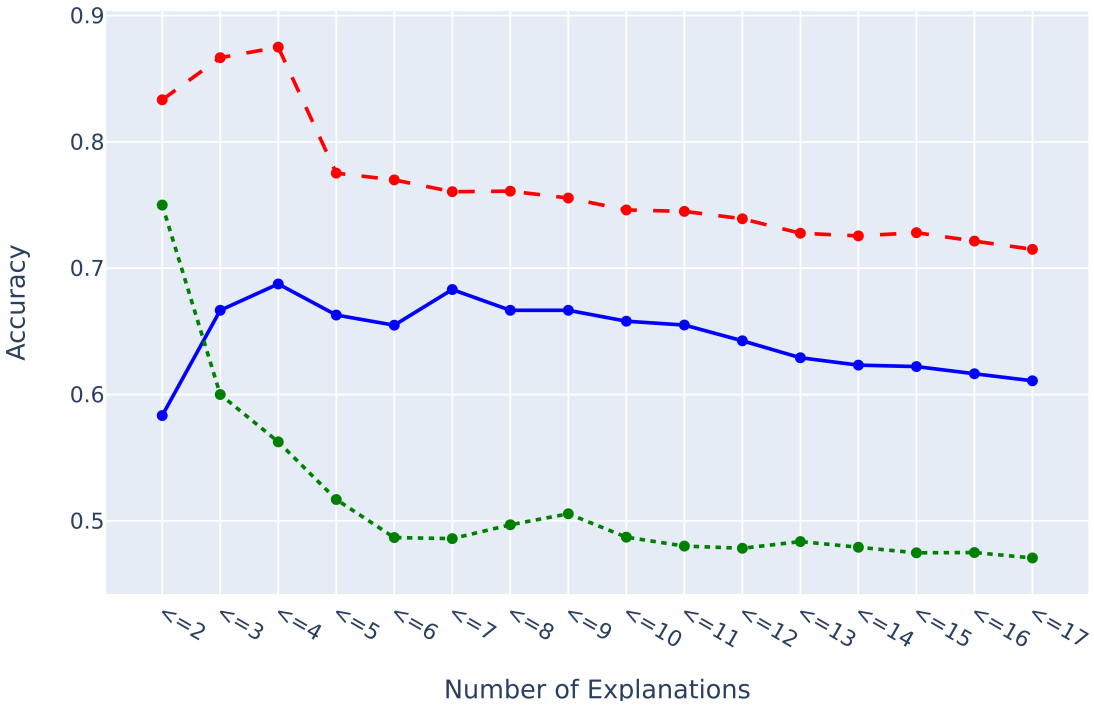

Figure 3.3: Change in accuracy of answer prediction the development set varying across different models with increasing explanation length for WorldTree *dev-set*. Red dashed line represents ExplanationLP + UR (k=30), blue line represents BERT$_{Large}$ + UR (k=10) and green dotted line represents PathNet + UR (k=20)

3.4 Related Work

Our approach broadly falls into Integer Linear Programming based approaches for science question answering. ILP-based approaches perform inference over either semi-structured tables (Khashabi et al., 2016) or structural representations extracted from the text (Khashabi, Khot, Sabharwal, & Roth, 2018; Khot et al., 2017). These approaches treat all facts homogeneously and attempt to connect the question with the correct answer through long hops. While they have exhibited good performance with no supervision, the performance tends to be lower when answering complex questions requiring long explanatory chains. In contrast, our approach performs inference over unstructured text by imposing structural constraints via grounding-abstract chains, lowering the hops, and combining parametric optimization to extract the best-performing model.

The other class of approaches that provide explanations are graph-based approaches. Graph-based approaches have been successfully applied for open-domain question answering (Fang et al., 2020; L. Qiu et al., 2019) where the question only requires only two hops. PathNet (Kundu et al., 2019) operates within the same design principles and

#	Approach	Accuracy	
		WT	**ARC**
1	ExplanationLP **(Best)**	**61.37**	**40.21**
Structure			
2	Grounding-Abstract Categories	58.33	35.13
3	Edge weights	43.78	29.45
4	Node weights	42.80	27.87
Cohesion			
5	Hypothesis-Abstract cohesion	38.71	30.37
6	Hypothesis-Grounding cohesion	59.33	38.73
7	Grounding-Abstract cohesion	59.12	38.14
Diversity			
8	Abstract-Abstract diversity	60.16	37.62
9	Grounding-Grounding diversity	60.44	37.71
Relevance			
10	Hypothesis-Abstract semantic similarity	55.38	35.49
11	Hypothesis-Abstract lexical relevance	54.68	36.01

Table 3.6: Ablation study, removing different components of ExplanationLP. The scores reported here are accuracy for answer selection on the WorldTree (WT) and ARC-Challenge (ARC) test-set.

has been applied to the OpenbookQA science dataset. As indicated in the empirical evaluation, it struggles with long-chain explanations since it relies only on two facts. Graph-based approaches have also been employed for mathematical reasoning (Ferreira & Freitas, 2020a, 2020b) and textual entailment (Silva et al., 2019; Silva et al., 2018).

The third category of partially explainable approaches employs black-box neural models in combination with a retrieval approach. The SOTA model for Science Question (Khashabi et al., 2020) answering is pretrained across multiple datasets and is not explainable. The current partially explainable SOTA approach that does not rely on external resource (Yadav et al., 2019b) employs a large parameter BERT model for question answering resulting. In contrast, with a low number of parameters, we have introduced a model that demonstrates competitive performance and leaves a smaller carbon footprint in terms of energy consumption (Henderson et al., 2020). Other methods construct explanation chains by leveraging explanatory patterns emerging in a corpus of scientific explanations (Valentino et al., 2020, 2021).

3.5 Conclusion

Research Objective 2:*Propose a novel explanation-based multi-hop inference method which reduces semantic drift*

This Chapter presented an efficient science question answering model that performs explanation-based inference. We also presented an in-depth systematic evaluation demonstrating the impact on various design principles via an in-depth ablation analysis. Despite having a significantly lower number of parameters, we demonstrated competitive performance compared with contemporary explanation-based approaches while showcasing its robustness and interpretability.

- **RQ2.1**: Does the encoding of grounding-abstract mechanisms reduce semantic drift?

  In Section 3.3.1, we demonstrated that our model outperforms Transformer-based models - BERT (Devlin et al., 2019) ($\approx 7\%$) and graph-based model-PathNet (Kundu et al., 2019) ($\approx 17\%$). In Section 3.3.4, we showed that ExplanationLP showed a lower degradation with an increasing number of explanation sentences. Demonstrating the positive impact of grounding-abstract mechanisms on the semantic drift. We also exhibit consistently better performance when compared with BERT as well. In Section 3.3.4 we also showed grounding-abstract chains play a significant role in answer selection performance (improvement by $\approx$ 5% for ARC-Challenge). With these observations, we can conclude that *encoding of grounding-abstract mechanisms reduce semantic drift*.

3.6 Scope and Limitations

As noted by Jansen et al. (2018) elementary science questions require an average of 4 facts to answer and explain, with some questions requiring over 20 pieces of information. However, empirically we found $M = 2$ of abstract facts to provide the best answer selection results. The central aim of the approach was to answer the question and not reconstruct the entire explanation chain. The impact of this method can be seen in the low F1 score in the explanation selection task. This design choice can also be attributed to the degradation of answer selection performance with increasing explanation sentences.

ExplanationLP relies on the existence of a corpus of core scientific statements (abstract facts). In our case, we were aided by the existence of WorldTree corpus (Jansen

et al., 2018).

As a framework, ExplanationLP is limited to multiple-choice question answering. If we were to adopt this approach to span-selection, one possible way is to convert span-selection questions into multiple-choice by extracting potential answer spans from the text (Du & Cardie, 2018).

This model was also limited to fine-tuning only nine parameters as it is intractable to fine-tune large models using Bayesian optimization.

3.7 Reproducibility

This section consists of all the hyperparameters, code and libraries used in our approach.

3.7.1 Integer Linear Programming Optimization

The components of the Integer Linear Programming system is as follows:

- Solver: CPLEX Solver - CPLEX optimization studio V12.9.0 https://www.ibm.com/products/ilog-cplex-optimization-studio

The hyperparameters used in the LP constraints:

- Maximum number of abstract facts (M): 2
- Average time per epoch: 6 minutes for train-set
- Number of Epochs: 200

Infrastructures used:

- CPU Cores: 32
- CPU Model: Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz
- Memory: 128GB
- OS: Ubuntu 18.04 LTS

3.7.2 Parameter tuning

Our work employed Bayesian optimization with a Gaussian process for hyperparameter tuning. We used the https://github.com/fmfn/BayesianOptimization: BayesianOptimization python library to implement the code. These parameters are as follows:

- Gaussian Kernels:
  - https://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.kernels.RationalQuadratic.html: RationalQuadratic Kernel with default parameters
- Number of iterations: 200
- alpha (α): 1e-8
- random state: 1

3.7.3 Sentence-BERT for Semantic Similarity Scores

We use: `roberta-large nli-stsb-mean-tokens` model to calculate the semantic similarity scores.

3.7.4 BERT model

The BERT model was taken from the Hugginface Transformers(https://github.com/huggingface/transformers) library and fine-tuned using 4 Tesla V100 GPUs for 10 epochs in total with batch size 16 for BERT_{Large} and 32 for BERT_{Base}.

We experiment with a range of hyperparameters and pick the hyperparameters with the best performance. The hyperparameter value range tested:

- learning rate: $\{1e-4,\ 5e-5,\ 1e-6\}$
- warmup steps : $\{0, 5,\ 10,\ 20\}$
- weight decay: $\{0.0,\ 1e-3,\ 1e-6\}$

The hyperparameters adopted for BERT are as follows:

- gradient accumulation steps: 1
- learning rate: 1e-5
- weight decay: 0.0
- adam epsilon: 1e-8
- warmup steps: 0
- max grad norm: 1.0
- seed: 42

3.7.5 PathNet

We use the code and dependencies provided by the PathNet GitHub repository (https://github.com/allenai/PathNet). We used the training config provided for OpenBookQA as

a baseline: https://github.com/allenai/PathNet/blob/master/training_configs/config_obqa.json.

3.7.6 Relevant facts retrieval

The code for BM25 and Unification retrieval approaches were adopted from the Unification Explanation Retrieval GitHub repository (https://github.com/ai-systems/unification_reconstruction_explanations).

3.7.7 Code

The code for reproducing the ExplanationLP and the experiments described in this chapter are attached with the code appendix and will be available at the following GitHub repository (with a Dockerized container): https://anonymous-url.com.

3.7.8 Data

WorldTree Dataset: The 2,290 questions in the WorldTree corpus are split into three different subsets: *train-set* (987), *dev-set* (226), and *test-set* (1,077). We only considered questions with explanations for our evaluation. The reasoning behind omitting questions without explanations was to ensure fact coverage for all questions. For AbstractKB building, we excluded facts from 'KINDOF' and 'SYNONYMY' table, as these are the ones primarily composed of grounding facts.
ARC-Challenge Dataset: https://allenai.org/data/arc. Only used the Challenge split.

Chapter 4

Diff-Explainer: Differentiable Convex Optimization for Explanation-based Multi-hop Inference

This Chapter is based on the paper "Diff-Explainer: Differentiable Convex Optimization for Explainable Multi-hop Inference". This can be found in https://arxiv.org/pdf/2105.03417.pdf and has been accepted for *Transactions of the Association for Computational Linguistics, 2022*.

4.1 Introduction

Explanation-based Question Answering (QA) in complex domains is often modelled as a *multi-hop inference* problem (Jansen et al., 2021; Valentino et al., 2022). In this context, the goal is to answer a given question through the construction of an explanation, typically represented as a graph of multiple interconnected sentences supporting the answer (Figure 4.1) (Jansen, 2018; Khashabi, Khot, Sabharwal, & Roth, 2018; Kundu et al., 2019).

However, Explanation-based QA models exhibit lower performance when compared to state-of-the-art approaches, which are generally represented by Transformer-based architectures (Devlin et al., 2019; Khashabi et al., 2020; Khot et al., 2020). While Transformers are able to achieve high accuracy due to their ability to transfer linguistic and semantic information to downstream tasks, they are typically regarded as black-boxes (Liang et al., 2021), posing concerns about the interpretability and transparency of their predictions (Guidotti et al., 2018; Rudin, 2019).

To alleviate the aforementioned limitations, this chapter proposes *Diff*-Explainer (∂-Explainer), a novel *hybrid* framework for multi-hop and explanation-based QA that combines constraint satisfaction layers with pre-trained neural representations, enabling end-to-end differentiability.

Recent works have shown that certain convex optimization problems can be represented as individual layers in larger end-to-end differentiable networks (Agrawal, Barratt, et al., 2019; Agrawal, Amos, et al., 2019; Amos & Kolter, 2017), demonstrating that these layers can be adapted to encode constraints and dependencies between hidden states that are hard to capture via standard neural networks.

In this chapter, we build upon this line of research, showing that convex optimization layers can be integrated with Transformers to improve explanation-based inference and robustness in multi-hop inference problems. To illustrate the impact of end-to-end differentiability, we integrate the constraints of existing ILP solvers (i.e., TupleILP (Khot et al., 2017), ExplanationLP) into a hybrid framework. Specifically, we propose a methodology to transform existing constraints into differentiable convex optimization layers and subsequently integrate them with pre-trained sentence embeddings based on Transformers (Reimers et al., 2019).

To evaluate the proposed framework, we perform extensive experiments on complex multiple-choice QA tasks requiring scientific and commonsense reasoning (Clark et al., 2018; Xie et al., 2020). In summary, the contributions of the chapter are as follows:

1. A novel differentiable framework for multi-hop inference that incorporates constraints via convex optimization layers into broader Transformer-based architectures.

2. An extensive empirical evaluation demonstrating that the proposed framework allows end-to-end differentiability on downstream QA tasks for both explanation and answer selection, leading to a substantial improvement when compared to non-differentiable constraint-based and transformer-based approaches.

3. We demonstrate that *Diff*-Explainer is more robust to distracting information in addressing multi-hop inference when compared to Transformer-based models.

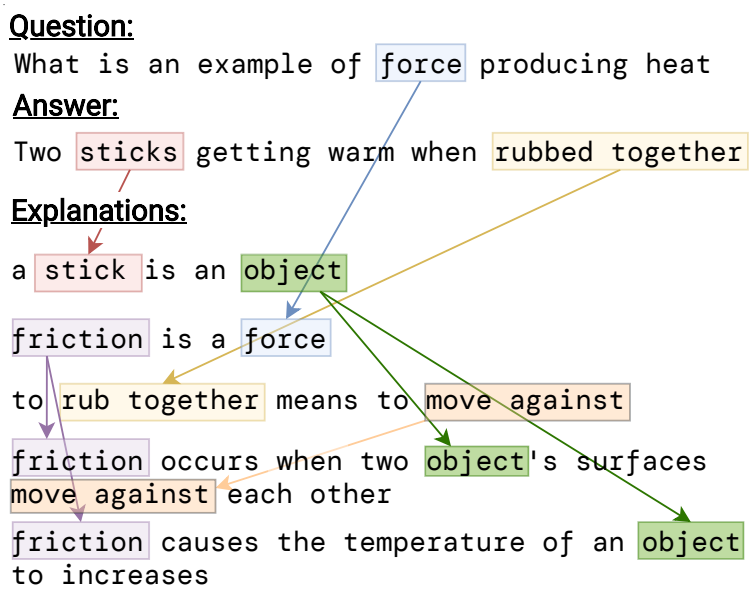

Figure 4.1: Example of a multi-hop QA problem with an explanation represented as a graph of multiple interconnected sentences supporting the answer (Jansen et al., 2018; Xie et al., 2020).

4.2 Differentiable Convex Optimization Layers

Our work is in line with previous works that have attempted to incorporate optimization as a neural network layer. These works have introduced differentiable modules for quadratic problems (Amos & Kolter, 2017; Donti et al., 2017), satisfiability solvers (P.-W. Wang et al., 2019) and submodular optimizations (Djolonga & Krause, 2017; Tschiatschek et al., 2018). Recent works also offer differentiation through convex cone programs (Agrawal, Barratt, et al., 2019; Busseti et al., 2019).

Given the primal(Equation 4.1)-dual(Equation 4.2) form of a cone program as follows:

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{s.t.} \quad & Ax + s = b \\ & x \in \mathcal{K} \end{aligned} \tag{4.1}$$

$$\begin{aligned} \text{minimize} \quad & b^T y \\ \text{s.t.} \quad & A^T y + c = 0 \\ & y \in \mathcal{K}^* \end{aligned} \tag{4.2}$$

Here

- $x \in \mathbb{R}^n$: *primal* variable
- $y \in \mathbb{R}^m$: *dual* variable
- $s \in \mathbb{R}^m$: primal *slack* variable

- $\mathcal{K} \subseteq \mathbb{R}^m$: nonempty, closed, convex cone
- $\mathcal{K}^* \subseteq \mathbb{R}^m$: dual cone of $\mathcal{K}$

Conic solver as a function can be seen as $\psi : \mathbb{R}^{m\times n} \times \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^{n+2m}$ mapping the problem data (A,b,c) to a solution (x,y,s).

ψ can be expressed as $\phi \circ s \circ Q$ (Agrawal, Barratt, et al., 2019; Amos, 2019), where

- $Q : \mathbb{R}^{m\times n} \times \mathbb{R}^m \times \mathbb{R}^n \to R^{N\times N}$ maps the problem data to Q, given by:

$$Q = \begin{bmatrix} 0 & A^T & c \\ -A & 0 & b \\ -c^T & -b^T & 0 \end{bmatrix}. \tag{4.3}$$

- $s : \mathbb{R}^{N\times N} \to \mathbb{R}^N$ is differentiable and solves the homogeneous self-dual embedding
- $\phi : \mathbb{R}^N \to \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^m$ maps z to the primal-dual pair, given by:

$$(x,y,s) = (u, \Pi_{\mathcal{K}^*}(v), \Pi_{\mathcal{K}^*}(v) - v)/w, \tag{4.4}$$

The adjoint of the derivative of ψ at (A,b,c) applied to the vector $(\mathrm{d}x, \mathrm{d}y, \mathrm{d}s)$, or

$$(\mathrm{d}A, \mathrm{d}b, \mathrm{d}c) = D^T\psi(A,b,c)(\mathrm{d}x,\mathrm{d}y,\mathrm{d}s) = D^T Q(A,b,c)D^T s(Q)D^T\phi(z)(\mathrm{d}x,\mathrm{d}y,\mathrm{d}s) \tag{4.5}$$

As the only interest is in the the primal solution x, they set $\mathrm{d}y = \mathrm{d}s = 0$. Hence based on Equation 4.4 they define:

$$\mathrm{d}z = D^T\phi(z)(\mathrm{d}x,0,0) = \begin{bmatrix} \mathrm{d}x \\ 0 \\ -x^T\mathrm{d}x \end{bmatrix} \tag{4.6}$$

$\mathrm{D}s(Q)$ is further calculated by implicitly differentiating the normalized residual map:

$$\mathrm{D}s(Q) = -(D_z\mathcal{N}(s(Q),Q))^{-1}D_Q\mathcal{N}(s(Q),Q). \tag{4.7}$$

Resulting in

$$\mathrm{d}Q = \mathrm{D}^T s(Q)\mathrm{d}z = -(M^{-T}\mathrm{d}z)\Pi(z)^T, \tag{4.8}$$

where $M = (Q-I)D\Pi(z)+I$. In order to calculate $g = M^{-T}\mathrm{d}z$ efficiently the LSQR Krylov method (Paige & Saunders, 1982) is used.

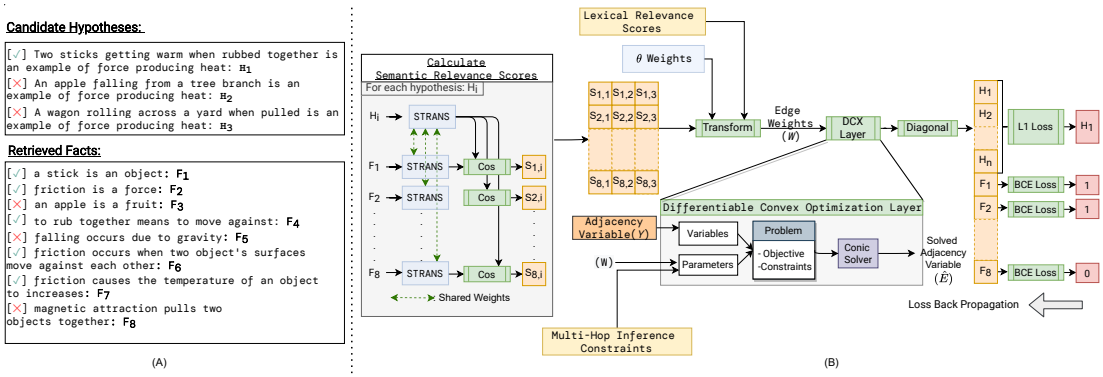

Figure 4.2: Overview of our approach: Illustrates the end-to-end architectural diagram of *Diff*-Explainer for the provided example.

dQ is calculated as

$$\mathrm{d}Q = \begin{bmatrix} \mathrm{d}Q_{11} & \mathrm{d}Q_{12} & \mathrm{d}Q_{13} \\ \mathrm{d}Q_{21} & \mathrm{d}Q_{22} & \mathrm{d}Q_{23} \\ \mathrm{d}Q_{31} & \mathrm{d}Q_{32} & \mathrm{d}Q_{33} \end{bmatrix}, \tag{4.9}$$

Finally, obtain

$$\begin{aligned} \mathrm{d}A = & \; -\mathrm{d}Q_{12}^T + \mathrm{d}Q_{21} \\ \mathrm{d}b = & \; -\mathrm{d}Q_{23} + \mathrm{d}Q_{32}^T \\ \mathrm{d}c = & \; -\mathrm{d}Q_{13} + \mathrm{d}Q_{31}^T \end{aligned} \tag{4.10}$$

Since every convex problem can be cast into a cone problem, these models can be used to define convex optimization problems. In this work, we use the differentiable convex optimization layers proposed by Agrawal, Amos, et al. (2019). These layers provide a way to abstract away from the conic form, letting users define convex optimization in natural syntax.

4.3 *Diff*-Explainer: Explanation-based Multi-Hop Inference via Differentiable Convex Optimization

The problem of Explanation-based Multi-Hop Question Answering can be stated as follows (Restated from Chapter 1):

Definition 3 (*Explanations in Multi-Hop Question Answering*)**.** Given a question Q, answer a and a knowledge base F_{kb} (composed of natural language sentences), we say that we may *infer* hypothesis h (where hypotheses h is the concatenation of Q with a) if there exists a subset (F_{exp}) of supporting facts $\{f_1, f_2, \ldots\} \subseteq F_{kb}$ of statements which

would allow to arrive at h from $\{f_1, f_2, \ldots\}$. We call this set of facts an *explanation* for h.

Given a question (Q) and a set of candidate answers $C = \{c_1,\ c_2,\ c_3, ...,\ c_n\}$ ILP-based approaches (Khashabi et al., 2016; Khot et al., 2017) convert them into a list of hypothesis $H = \{h_1,\ h_2,\ h_3,\ \ldots,\ h_n\}$ by concatenating question and candidate answer. For each hypothesis h_i these approaches typically adopt a retrieval model (e.g: BM25, FAISS (Johnson et al., 2019)), to select a list of candidate explanatory facts $F = \{f_1,\ f_2,\ f_3,\ \ldots,\ f_k\}$, and construct a weighted graph $G = (V, E, W)$ with edge weights $W : E \rightarrow \mathbb{R}$ where $V = \{\{h_i\} \cup F\}$, edge weight W_{ik} of each edge E_{ik} denote how relevant a fact f_k is with respect to the hypothesis h_i.

Based on these definitions, ILP-based QA can be defined as follows (Restated from Chapter 1):

Definition 4 (*ILP-Based Multi-Hop QA*)**.** Find a subset $\tilde{V} \subseteq V$, $h \in \tilde{V}$, $\tilde{V} \setminus \{h\} = F_{exp}$ and $\tilde{E} \subseteq E$ such that the induced subgraph $\tilde{G} = (\tilde{V}, \tilde{E})$ is connected, weight $W[\tilde{G} = (\tilde{V}, \tilde{E})] := \sum_{e \in \tilde{E}} W(e)$ is maximal and adheres to set of constraints M_c designed to emulate multi-hop inference. The hypothesis h_i with the highest subgraph weight $W[\tilde{G} = (\tilde{V},\ \tilde{E})]$ is selected to be the correct answer c_{ans}.

The ILP-based inference has two main challenges in producing convincing explanations. First, design edge weights W, ideally capturing a quantification of the relevance of the fact to the hypothesis. Second, define constraints that emulate the multi-hop inference process.

4.3.1 Limitations with Existing ILP formulations

In previous work, the construction of the graph G requires predetermined edge-weights based on lexical overlaps (Khot et al., 2017) or semantic similarity using sentence embeddings, on top of which combinatorial optimization strategies are performed separately. From those approaches, ExplanationLP proposed in Chapter 3 is the only approach that modifies the graph weight function by optimizing the weight parameters θ by fine-tuning them for inference via Bayesian Optimization over pre-trained embeddings.

In contrast, we posit that learning the graph weights dynamically by fine-tuning the underlying neural embeddings towards answer and explanation selection will lead to more accurate and robust performance. To this end, the constraint optimization strategy

should be differentiable and efficient. However, Integer Linear Programming based approaches present two critical shortcomings that prevent achieving this goal:

1. The Integer Linear Programming formulation operates with discrete inputs/outputs resulting in *non-differentiability* (Paulus et al., 2021). Consequently, it cannot be integrated with deep neural networks and trained end-to-end. Making ILP differentiable requires non-trivial assumptions and approximations (Paulus et al., 2021).
2. Integer Programming is known to be NP-complete, with the special case of 0-1 integer linear programming being one of Karp's 21 NP-complete problems (Karp, 1972). Therefore, as the size of the combinatorial optimization problem increases, finding exact solutions becomes computationally intractable. This intractability is a strong limitation for multi-hop QA in general since these systems typically operate on large knowledge bases and corpora.

4.3.2 Subgraph Selection via Semi-Definite Programming

Differentiable convex optimization (DCX) layers (Agrawal, Amos, et al., 2019) provide a way to encode constraints as part of a deep neural network. However, an ILP formulation is non-convex (Schrijver, 1998; Wolsey, 2020) and cannot be incorporated into a differentiable convex optimization layer. The challenge is to approximate ILP with convex optimization constraints.

In order to alleviate this problem, we turn to *Semi-Definite programming* (SDP) (Vandenberghe & Boyd, 1996). SDP is non-linear but convex and has shown to efficiently approximate combinatorial problems.

A semi-definite optimization is a convex optimization of the form:

$$\begin{aligned} &\textit{minimize} && C \cdot X && (4.11)\\ &\textit{s.t} && A \cdot X = b_i,\ i = 1,2,\ \cdots,\ m, && (4.12)\\ & && X \succeq 0, && (4.13) \end{aligned}$$

Here $X \in \mathbb{S}^n$ is the optimization variable and $C, A_1, \ldots, A_p \in \mathbb{S}^n$, and $b_1, \ldots, b_p \in \mathbb{R}$. $X \succeq 0$ is a matrix inequality with $\mathbb{S}^n$ denotes a set of $n \times n$ symmetric matrices.

SDP is often used as a convex approximation of traditional NP-hard combinatorial graph optimization problems, such as the max-cut problem, the dense k-subgraph problem and the quadratic $\{0-1\}$ programming problem (Lovász & Schrijver, 1991).

Specifically, we adopt the semi-definite relaxation of the following quadratic $\{0,\ 1\}$ problem:

$$\textit{maximize} \qquad y^T W y \tag{4.14}$$

$$y \in \{0,1\}^n \tag{4.15}$$

Here W is the edge weight matrix of the graph G and the optimal solution for this problem $\hat{y}$ indicates if a node is part of the induced subgraph $\tilde{G}$.

We follow Helmberg (2000) in their reformulation and relaxation of this problem. Instead of vectors $y \in \{0,1\}^n$, we optimize over the set of *positive semidefinite matrices* satisfying the SDP constraint in the following relaxed convex optimization problem[1]:

$$\textit{maximize} \qquad \langle W, Y \rangle \tag{4.16}$$

$$\textit{s.t} \qquad Y - diag(Y)diag(Y)^T \succeq 0 \tag{4.17}$$

where $\langle W,Y\rangle = \text{trace}(WY)$, $Y = yy^T$, $diag(Y) = y$.

The optimal solution for Y in this problem $\hat{E} \in [0,\ 1]$ indicates if an edge is part of the subgraph $\tilde{G}$. In addition to the semi-definite constraints, we impose Multi-hop inference constraints M_c. These constraints are introduced in Section 4.3.6 and the Appendix.

This reformulation provides the tightest approximation for the optimization with the convex constraints. Since this formulation is convex, we can now integrate it with differentiable convex optimization layers. Moreover, the semi-definite program relaxation can be solved by adopting the interior-point method (De Klerk, 2006; Vandenberghe & Boyd, 1996) which has been proved to run in polynomial time (Karmarkar, 1984). To the best of our knowledge, we are the first to employ SDP to solve a natural language processing task.

4.3.3 *Diff*-Explainer: End-to-End Differentiable Architecture

Diff-Explainer is an end-to-end differentiable architecture that simultaneously solves the constraint optimization problem and dynamically adjusts the graph edge weights for better performance. We adopt *differentiable convex optimization* for the optimal

[1] See (Helmberg, 2000) for the derivation from the original optimization problem.

subgraph selection problem. The complete architecture and setup are described in the subsequent subsections and Figure 4.2.

We transform a multi-hop question answering dataset into a multi-hop QA dataset by converting an example's question (q) and the set of candidate answers $C = \{c_1,\ c_2,\ c_3,\ \ldots,\ c_n\}$ into hypotheses $H = \{h_1,\ h_2,\ h_3,\ \ldots,\ h_n\}$ (See Figure 4.2A) by using the approach proposed by Demszky et al. (2018). To build the initial graph, for the hypotheses set H we adopt a retrieval model to select a list of candidate explanatory facts $F = \{f_1,\ f_2,\ f_3,\ \ldots,\ f_k\}$ to construct a weighted complete bipartite graph $G = (H,\ F,\ E,\ W)$, where the weights W_{ik} of each edge E_{ik} denote how relevant a fact f_k is with respect to the hypothesis h_i. Departing from traditional ILP approaches (Khashabi et al., 2016; Khashabi, Khot, Sabharwal, & Roth, 2018), the aim is to select the correct answer c_{ans} and relevant explanations F_{exp} with a single graph.

In order to demonstrate the impact of *Diff*-Explainer, we reproduce the formalization introduced by previous ILP solvers. Specifically, we approximate the two following solvers:

- **TupleILP** (Khot et al., 2017): TupleILP constructs a semi-structured knowledge base using tuples extracted via Open Information Extraction (OIE) and performs inference over them. TupleILP uses Subject-Predicate-Object tuples for aligning and constructing the explanation graph. As shown in Figure 4.3C, the tuple graph is constructed and lexical overlaps are aligned to select the explanatory facts. The constraints are designed based on the position of text in the tuple.
- **ExplanationLP**: ExplanationLP classifies facts into abstract and grounding facts. Abstract facts are core scientific statements. Grounding facts help connect the generic terms in the abstract facts to the terms in the hypothesis. For example, in Figure 4.2A, F_1 is a grounding fact and helps to connect the hypothesis with the abstract fact F_7. The approach aims to emulate abstract reasoning.

We have also presented Figure 4.3B for reference to show how ExplanationLP acts compared to TupleILP.

To demonstrate the impact of integrating a convex optimization layer into a broader end-to-end neural architecture, *Diff*-Explainer employs a transformer-based sentence embedding model. Figure 4.2B describes the end-to-end architectural diagram of *Diff*-Explainer. Specifically, we incorporate a differentiable convex optimization layer with Sentence-Transformer (STrans) (Reimers et al., 2019), which has demonstrated state-of-the-art performance on semantic sentence similarity benchmarks.

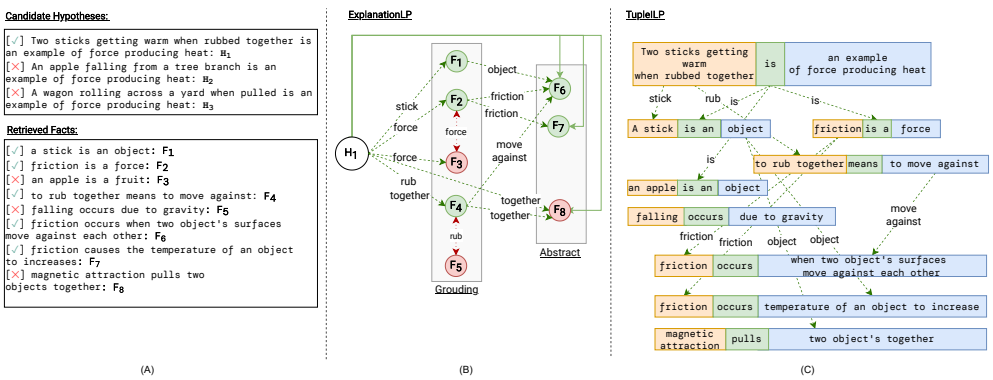

Figure 4.3: ILP-Based Multi-hop Inference.

STrans is adopted to estimate the relevance between hypothesis and facts during the construction of the initial graph. We use STrans as a bi-encoder architecture to minimize the computational overload and operate on a large number of sentences. The semantic relevance score from STrans is complemented with a lexical relevance score computed considering the shared terms between hypotheses and facts. We calculate semantic and lexical relevance as follows:

Semantic Relevance (s): Given a hypothesis h_i and fact f_j we compute sentence vectors of $\vec{h}_i = STrans(h_i)$ and $\vec{f}_j = STrans(f_j)$ and calculate the semantic relevance score using cosine-similarity as follows:

$$s_{ij} = S(\vec{h}_i,\ \vec{f}_j) = \frac{\vec{h}_i \cdot \vec{f}_j}{||\vec{h}_i||||\vec{f}_j||} \tag{4.18}$$

Lexical Relevance (l): The lexical relevance score of hypothesis h_i and f_j is given by the percentage of overlaps between unique terms (here, the function trm extracts the lemmatized set of unique terms from the given text):

$$l_{ij} = L(h_i,\ f_j) = \frac{|trm(h_i) \cap trm(f_j)|}{max(|trm(h_i)|, |trm(f_j)|)} \tag{4.19}$$

Given the above scoring function, we construct edge weights matrix (W) as follows:

$$W_{ij} = [\theta^s_1,\ \theta^s_2,\ \ldots, \theta^s_n] \cdot [s^{\mathcal{D}_1}_{ij},\ s^{\mathcal{D}_2}_{ij},\ \ldots, s^{\mathcal{D}_n}_{ij}] + [\theta^l_1,\ \theta^l_2,\ \ldots, \theta^l_n] \cdot [l^{\mathcal{D}_1}_{ij},\ l^{\mathcal{D}_2}_{ij},\ \ldots, l^{\mathcal{D}_n}_{ij}] \tag{4.20}$$

Here relevance scores are weighted by weight parameters (θ) with θ clamped to $[0, 1]$. $s^{\mathcal{D}_k}_{ij}$ (or $l^{\mathcal{D}_k}_{ij}$) is s_{ij} (or l_{ij}) if (i, j) satisfy condition $\mathcal{D}_k$ or 0 otherwise.

4.3.4 Objective Function

In this section, we explain how to design the objective function for TupleILP and ExplanationLP to adopt with *Diff*-Explainer.

Given n candidate hypotheses and k candidate explanatory facts, A represents an adjacency matrix of dimension $((n+k)\times(n+k))$ where the first n columns and rows denote the candidate hypotheses, while the remaining rows and columns represent the candidate explanatory facts. The adjacency matrix denotes the graph's lexical connections between hypotheses and facts. Specifically, each entry in the matrix A_{ij} contains the following values:

$$Aij = \begin{cases} 1, & i \leq n, j > n, \ |trm(h_i) \cap trm(f_{j-n})| > 0 \\ 1, & j \leq n, i > n, \ |trm(h_j) \cap trm(f_{i-n})| > 0 \\ 0, & \text{otherwise} \end{cases} \tag{4.21}$$

Given the relevance scoring functions, we construct edge weights matrix (W) via a weighted function for each approach as follows:

TupleILP The weight function for *Diff*-Explainer with TupleILP constraints is:

$$W_{ij} = (\theta_{s_r} S_{ij} + \theta_{l_r} L_{ij}) \times A_{ij} \ \forall i, j \in V \tag{4.22}$$

ExplanationLP Give Abstract KB (F_A) and Grounding KB (F_G), the weight function for *Diff*-Explainer with Explanation LP is as follows:

$$W_{ij} = \begin{cases} -\theta_{gg} L_{ij} & v_j, v_k \in F_G \\ -\theta_{aa} L_{ij} & v_j, v_k \in F_A \\ \theta_{ga} L_{ij} & v_j \in F_G, v_k \in F_A \\ \theta_{qgl} L_{ij} + \theta_{qgs} S_{ij} & v_j \in F_G, v_k = h_i \\ \theta_{qal} L_{ij} + \theta_{qal} S_{ij} & v_j \in F_A, v_k = h_i \end{cases} \tag{4.23}$$

4.3.5 Constraints with Disciplined Parameterized Programming (DPP)

In order to adopt differentiable convex optimization layers, the constraints should be defined following the Disciplined Parameterized Programming (DPP) formalism (Agrawal,

Amos, et al., 2019), providing a set of conventions when constructing convex optimization problems. DPP consists of functions (or *atoms*) with a known curvature (affine, convex or concave) and per-argument monotonicities. In addition to these, DPP also consists of *Parameters* which are symbolic constants with an unknown numerical value assigned during the solver run.

TupleILP We extract SPO tuples $f_i^t = \{f_i^S,\ f_i^P,\ f_i^O\}$ for each fact f_i using an Open Information Extraction model (Stanovsky et al., 2018). From the hypothesis h_i we extract the set of unique terms $h_i^{ht} = \{t_1^{h_i},\ t_2^{h_i},\ t_3^{h_i},\ \ldots,\ t_l^{h_i}\}$ excluding stopwords.

In addition to the aforementioned constraints and semidefinite constraints specified in Equation 4.17, we adopt part of the constraints from TupleILP (Khot et al., 2017). In order to implement TupleILP constraints, we extract SPO tuples $f_i^t = \{f_i^S,\ f_i^P,\ f_i^O\}$ for each fact f_i using an Open Information Extraction model (Stanovsky et al., 2018). From the hypotheses H we also extract the set of unique terms $H^t = \{t_1,\ t_2,\ t_3,\ \ldots,\ t_l\}$ excluding stopwords. The constraints are described in Table 4.1.

ExplanationLP ExplanationLP constraints are described in Table 4.1.

4.3.6 Answer and Explanation Selection

Given edge variable Y and node variable y ($diag(Y) = y$) (See Section 4.3.2) where 1 means the edge/node is part of the subgraph and 0 otherwise, we design the the answer selection constraint is defined as follows:

$$\sum_{i \in H} Y_{ii} = 1 \tag{4.24}$$

Each entry in the edge diagonal represents a value between 0 and 1, indicating whether the corresponding node in the initial graph should be included in the optimal subgraph.

Explanation selection is done via the following constraint that limits the number of nodes in the subgraph to be m.

$$\sum_{i \in V} Y_{ii} = m + 1 \tag{4.25}$$

Besides these functional constraints, ILP-based methods also impose semantic and structural constraints. For instance, ExplanationLP places explicit grounding-abstract fact chain constraints to perform efficient abstractive reasoning and TupleILP enforces

constraints to leverage the SPO structure to align and select facts. See the Appendix on how these constraints are designed and imposed within *Diff*-Explainer.

The output from the DCX layer returns the solved edge adjacency matrix $\hat{E}$ with values between 0 and 1. We interpret the diagonal values of $\hat{E}$ as the probability of the specific node being part of the selected subgraph. The final step is to optimize the sum of the L1 loss l_1 between the selected answer and the correct answer c_{ans} for the answer loss $\mathcal{L}_{ans}$:

$$\mathcal{L}_{ans} = l_1(diag(\hat{E})[h_1,\ h_2,\ \ldots,\ h_n],\ c_{ans}) \tag{4.26}$$

As well as the binary cross entropy loss l_b between the selected explanatory facts and true explanatory facts F_{exp} for the explanatory loss $\mathcal{L}_{exp}$:

$$\mathcal{L}_{exp} = l_b(diag(\hat{E})[f_1,\ f_2,\ \ldots,\ f_k],\ F_{exp}) \tag{4.27}$$

We add the losses to backpropagate to learn the θ weights and fine-tune the sentence transformers. The pseudo-code to train *Diff*-Explainer end-to-end is summarized in Algorithm 3.

Algorithm 3: Training *Diff*-Explainer.

Data: $M_c \leftarrow$ Multi-hop inference constraints
Data: $Ans_c \leftarrow$ Answer selection constraint
Data: $Exp_c \leftarrow$ Explanation selection constraint
Data: $f_w \leftarrow$ Graph weight function
$G \leftarrow$ fact-graph-construction$(H,\ F)$;
$l \leftarrow L(H,\ F)$;
$\theta \leftarrow clamp([0,\ 1])$;
$epoch \leftarrow 0$;
while $epoch \leq max\_epochs$ **do**
  $\vec{F} \leftarrow STrans(F)$;
  $\vec{H} \leftarrow STrans(H)$;
  $s \leftarrow S(\vec{H},\ \vec{F})$;
  $W \leftarrow f_w(s,\ l;\ \theta)$;
  $\hat{E} \leftarrow DCX(W,\ \{M_c,\ Ans_c,\ Exp_c\})$;
  $\hat{V} \leftarrow diag(\hat{E})$;
  $\mathcal{L}_{ans} \leftarrow l_1(\hat{V}[h_1,\ h_2,\ \ldots,\ h_n],\ c_{ans})$;
  if F_{exp} *is available* **then**
    $\mathcal{L}_{exp} \leftarrow l_b(\hat{V}[f_1,\ f_2,\ \ldots,\ f_k],\ F_{exp})$;
    $loss = \mathcal{L}_{ans} + \mathcal{L}_{exp}$;
  else
    $loss = \mathcal{L}_{ans}$;
  end
  update θ, *STrans* using AdamW optimizer by minimizing *loss*;
  $epoch \leftarrow epoch + 1$;
end
Result: Store best θ and *STrans*

Description	DPP Format	Parameters
TupleILP		
Sub graph must have $\leq w_1$ active tuples	$\sum_{i \in F} E_{ii} \leq w_1 + 1$	-
Active hypothesis term must have $\leq w_2$ edges	$H_\theta[:,:,i] \odot E \leq w_2 \quad \forall i \in H^t$	H_θ is populated by hypothesis term matrix H with dimension $((n+k) \times (n+k) \times l)$ and the values are given by: $H_{ijk} = \begin{cases} 1, & \forall\, k \in H^t, i \in H, j \in F, \\ & t_k \in trm(h_i), t_k \in trm(f_j) \\ 1, & \forall\, k \in H^t, i \in F, j \in H, \\ & t_k \in trm(h_j), t_k \in trm(f_i) \\ 0, & \text{otherwise} \end{cases}$ (4.28)
Active tuple must have active subject	$E \odot T^S_\theta >= E \odot A_\theta$	A_θ populated by adjacency matrix A, T^S_θ by subject tuple matrix T^S with dimension $((n+k) \times (n+k))$ and the values are given by: $T^S_{ij} = \begin{cases} 1, & i \in H,\ j \in F, \\ & \lvert trm(h_i) \cap trm(f^S_j) \rvert > 0 \\ 1, & i \in F,\ j \in H, \\ & \lvert trm(h_j) \cap trm(f^S_i) \rvert > 0 \\ 0, & \text{otherwise} \end{cases}$ (4.29)
Active tuple must have $\geq w_3$ active fields	$E \odot T^S_\theta + E \odot T^P_\theta + E \odot T^O_\theta \geq w_3(E \odot A_\theta)$	A_θ populated by adjacency matrix A and T^S_θ, T^P_θ, T^O_θ populated by subject, predicate and object matrices T^S, T^P, T^O respectively. Predicate and object tuples are converted into T^P, T^O matrices similar to T^S
Active tuple must have an edge to some hypothesis term	Implemented during graph construction by only considering tuples that have lexical overlap with a hypothesis	-
ExplanationLP		
Limits the total number of abstract facts to w_4	$diag(E) \cdot F^{AB}_\theta \leq w_4$	F^{AB}_θ is populated by Abstract fact matrix F^{AB}, where: $F^{AB}_{ij} = \begin{cases} 1, & i \in H,\ j \in F^A \\ 0, & \text{otherwise} \end{cases}$ (4.30)

Table 4.1: Adopting TupleILP and ExplanationLP constraints in DPP format. For this work we set the hyperparameters w_1=2, w_2=2, w_3=1 and w_4=2.

4.4 Empirical Evaluation

Question Sets: We use the following multiple-choice question sets to evaluate the *Diff*-Explainer.

1. **WorldTree Corpus** (Xie et al., 2020): The 4,400 question and explanations in the WorldTree corpus are split into three different subsets: *train-set*, *dev-set* and *test-set*. We use the *dev-set* to assess the explanation selection performance since the explanations for *test-set* are not publicly available.

2. **ARC-Challenge Corpus** (Clark et al., 2018): ARC-Challenge is a multiple-choice question dataset which consists of question from science exams from grade 3 to grade 9. These questions have proven to be challenging to answer for other LP-based question answering and neural approaches.

Experimental Setup : We use *all-mpnet-base-v2* model as the Sentence Transformer model for the sentence representation in *Diff*-Explainer. The motivation to choose this model is to use a pre-trained model on natural language inference and MPNet$_{Base}$ (Song et al., 2020) is smaller compared to large models like BERT$_{Large}$, enabling us to encode a larger number of facts. Similarly, for fact retrieval representation, we use *all-mpnet-base-v2* trained with gold explanations of WorldTree Corpus to achieve a Mean Average Precision of 40.11 in the dev-set. We cache all the facts from the background knowledge and retrieve the top k facts using MIPS retrieval (Johnson et al., 2019). We follow a similar setting used in Chapter 3 for the background knowledge base by combining over 5000 abstract facts from the WorldTree table store (WTree) and over 100,000 *is-a* grounding facts from ConceptNet (CNet) (Speer et al., 2017). Furthermore, we also set m=2 in line with the previous configurations from TupleILP and ExplanationLP.

Baselines: In order to assess the complexity of the task and the potential benefits of the convex optimization layers presented in our approach, we show the results for different baselines. We run all models with $k = \{1, \ldots, 10, 25, 50, 75, 100\}$ to find the optimal setting for each baseline and perform a fair comparison. For each question, the baselines take as input a set of hypotheses, where each hypothesis is associated with k facts, ranked according to the fact retrieval model.

1. **IR Solver** (Clark et al., 2018): This approach attempts to answer the questions by computing the accumulated score from all k obtained from summing up the

retrieval scores. In this case, the retrieval scores are calculated using the cosine similarity of fact and hypothesis sentence vectors obtained from the STrans model trained on gold explanations. The hypothesis associated with the highest score is selected as the one containing the correct answer.

2. **BERT$_{Base}$ and BERT$_{Large}$** (Devlin et al., 2019): To use BERT for this task, we concatenate every hypothesis with k retrieved facts, using the separator token `[SEP]`. We use the HuggingFace (Wolf et al., 2019) implementation of *BertForSequenceClassification*, taking the prediction with the highest probability for the positive class as the correct answer.
3. **PathNet** (Kundu et al., 2019): PathNet is a graph-based neural approach that constructs a single linear path composed of two facts connected via entity pairs for reasoning. It uses the constructed paths as evidence of its reasoning process. They have exhibited strong performance for multiple-choice science questions.
4. **TupleILP** and **ExplanationLP**: Both replications of the non-differentiable solvers are implemented with the same constraints as *Diff*-Explainer via SDP approximation without fine-tuning end-to-end; instead, we fine-tune the θ parameters using Bayesian optimization and frozen STrans representations. This baseline helps us to understand the impact of the end-to-end fine-tuning.

4.4.1 Answer Selection

WorldTree Corpus: Table 4.2 presents the answer selection performance on the WorldTree corpus in terms of accuracy, presenting the best results obtained for each model after testing for different values of k. We also include the results for BERT without explanation in order to evaluate the influence extra facts can have on the final score. We also present the results for two different training goals, optimizing for only the answer and optimizing jointly for answer and explanation selection.

We draw the following conclusions from the empirical results obtained on the WorldTree corpus (The performance increase here are in expressed absolute terms):

1. *Diff*-Explainer with ExplanationLP and TupleILP outperforms the respective non-differentiable solvers by 13.3% and 8.91%. This increase in performance indicates that *Diff*-Explainer can incorporate different types of constraints and significantly improve performance compared with the non-differentiable version.

2. It is evident from the performance obtained by a large model such as BERT$_{Large}$ (59.32%) that we are dealing with a non-trivial task. The best *Diff*-Explainer

Model	Acc
Baselines	
IR Solver	50.48
BERT_{Base} (Without Retrieval)	45.43
BERT_{Base}	58.06
BERT_{Large} (Without Retrieval)	49.63
BERT_{Large}	59.32
TupleILP	49.81
ExplanationLP	62.57
PathNet	43.40
***Diff*-Explainer**	
TupleILP constraints	
- Answer Selection only	61.13
- Answer and explanation selection	63.11
ExplanationLP constraints	
- Answer selection only	69.73
- Answer and explanation selection	**71.48**

Table 4.2: Answer selection performance for the baselines and across different configurations of our approach on WorldTree Corpus.

setting (with ExplanationLP) outperforms the best transformer-based models with and without explanations by 12.16% and 21.85%. Additionally, we can also observe that both with TupleILP and ExplanationLP, we obtain better scores over the transformer-based configurations.

3. Fine-tuning with explanations yielded better performance than only answer selection with ExplanationLP and TupleILP, improving performance by 1.75% and 1.98%. The increase in performance indicates that *Diff*-Explainer can learn from the distant supervision of answer selection and improve in a strong supervision setting.

4. Overall, we can conclude that incorporating constraints using differentiable convex optimization with transformers for multi-hop QA leads to better performance than pure constraint-based or transformer-only approaches.

ARC Corpus: Table 4.3 presents a comparison of baselines and our approach with different background knowledge bases: TupleInf, the same as used by TupleILP (Khot

Model	Background KB	Acc
TupleILP (Khot et al., 2017)	TupleInf	23.83
ExplanationLP	WTree & CNet	40.21
TupleILP (Ours)	TupleInf	29.12
ExplanationLP (Ours)	WTree & CNet	37.40
***Diff*-Explainer**		
TupleILP Constraints	TupleInf	33.95
ExplanationLP Constraints	WTree & CNet	**42.95**

Table 4.3: Answer Selection performance on ARC corpus with *Diff*-Explainer fine-tuned on answer selection.

et al., 2017), and WorldTree & ConceptNet as used by ExplanationLP. We have also reported the original scores reported by the respective approaches.

For this dataset, we use our approach with the same settings as the model applied to WorldTree, and fine-tune for only answer selection since ARC does not have gold explanations. Models employing Large Language Models (LLMs) trained across multiple question answering datasets like UnifiedQA (Khashabi et al., 2020) and AristoBERT (Xu et al., 2021) have demonstrated strong performance in ARC with an accuracy of 81.14 and 68.95 respectively.

To ensure a fair comparison, we only compare the best configuration of *Diff*-Explainer with other approaches that have been trained *only* on the ARC corpus and provide some form of explanations in Table 4.4. Here the explainability column indicates if the model delivers an explanation for the predicted answer. A subset of the approaches produces evidence for the answer but remains intrinsically black-box. These models have been marked as *Partial*.

1. *Diff*-Explainer improves the performance of non-differentiable solvers regardless of the background knowledge and constraints. With the same background knowledge, our model improves the original TupleILP and ExplanationLP by 10.12% and 2.74%, respectively.

2. Our approach also achieves the highest performance for partially and fully explainable approaches trained *only* on ARC corpus.

3. As illustrated in Table 4.4, we outperform the next best fully explainable baseline (ExplanationLP) by 2.74%. We also outperform the state-of-the-art model

Model	Explainable	Accuracy
BERT_{Large}	No	35.11
IR Solver (Clark et al., 2016)	Yes	20.26
TupleILP (Khot et al., 2017)	Yes	23.83
TableILP (Khashabi et al., 2016)	Yes	26.97
ExplanationLP	Yes	40.21
DGEM (Clark et al., 2016)	Partial	27.11
KG^2 (Y. Zhang et al., 2018)	Partial	31.70
ET-RR (Ni et al., 2019)	Partial	36.61
Unsupervised AHE (Yadav et al., 2019a)	Partial	33.87
Supervised AHE (Yadav et al., 2019a)	Partial	34.47
AutoRocc (Yadav et al., 2019b)	Partial	**41.24**
Diff-Explainer (ExplanationLP)	Yes	**42.95**

Table 4.4: ARC challenge scores compared with other Fully or Partially explainable approaches trained *only* on the ARC dataset.

AutoRocc (Yadav et al., 2019b) (uses BERT_{Large}) that is only trained on ARC corpus by 1.71% with 230 million fewer parameters.

4. Overall, we achieve consistent performance improvement over different knowledge bases (TupleInf, Wordtree & ConceptNet) and question sets (ARC, WorldTree), indicating the robustness of the approach.

4.4.2 Explanation Selection

Table 4.5 shows the Precision@K scores for explanation retrieval for PathNet, ExplanationLP/TupleILP and *Diff*-Explainer with ExplanationLP/TupleILP trained on answer and explanation selection. We choose Precision@K as the evaluation metric as the design of the approaches is not to construct full explanations but to take the top k=2 explanations and select the answer.

As evident from the table, our approach significantly outperforms PathNet. We also

Model	Precision@1	Precision@2
TupleILP	40.44	31.21
ExplanationLP	51.99	40.41
PathNet	19.79	13.73
***Diff*-Explainer**		
TupleILP (Best)	40.64	32.23
ExplanationLP (Best)	**56.77**	**41.91**

Table 4.5: F1 score for explanation selection in WorldTree *dev*-set .

improved the explanation selection performance over the non-differentiable solvers indicating the end-to-end fine-tuning also helps improve the selection of explanatory facts.

4.4.3 Answer Selection with Increasing Distractors

As noted by previous works (Yadav et al., 2019b, 2020), the answer selection performance can decrease when increasing the number of used facts k for Transformer. We evaluate how our approach stacks compared with transformer-based approaches in this aspect, presented in Figure 4.4. As we can see, the IR Solver decreases in performance as we add more facts, while the scores for transformer-based models start deteriorating for $k > 5$. Such results might seem counter-intuitive since it would be natural to expect a model's performance to increase as we add supporting facts. However, in practice, that does not apply as by adding more facts, there is an addition of distractors that such models may not filter out.

We can prominently see this for BERT$_{Large}$ with a sudden drop in performance for $k = 10$, going from 56.61 to 30.26. Such a drop is likely caused by substantial overfitting; with the added noise, the model partially lost the ability for generalization. A softer version of this phenomenon is also observed for BERT$_{Base}$.

In contrast, our model's performance increases as we add more facts, reaching a stable point around $k = 50$. Such performance stems from our overlap and relevance scores and structural and semantic constraints. The obtained results highlight our model's robustness to distracting knowledge, allowing its use in data-rich scenarios, where one needs to use facts from extensive knowledge bases. PathNet is also exhibiting robustness across increasing distractors, but we consistently outperform it across all k

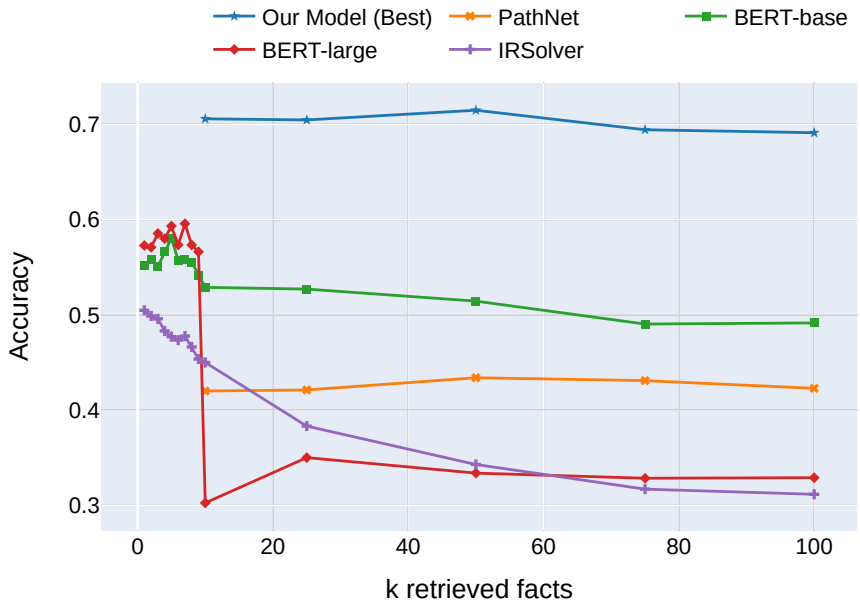

Figure 4.4: Comparison of accuracy for different number of retrieved facts.

configurations.

On the other hand, for smaller values of k our model is outperformed by transformer-based approaches, hinting that our model is more suitable for scenarios involving large knowledge bases such as the one presented in this work.

4.4.4 Qualitative Analysis

We selected some qualitative examples that showcase how end-to-end fine-tuning can improve the quality and inference and presented them in Table 4.6. We use the ExplanationLP for non-differentiable solver and *Diff*-Explainer as they yield higher performance in answer and explanation selection.

For Question (1), *Diff*-Explainer retrieves both explanations correctly and be able to answer correctly. Both PathNet and ExplanationLP has correctly retrieved at least one explanation but performed incorrect inference. We hypothesize that the other two approaches were distracted by the lexical overlaps in question/answer and facts, while our approach is robust towards distractor terms. In Question (2), our model was able only to retrieve one explanation correctly and was distracted by the lexical overlap to retrieve an irrelevant one. However, it still was able to answer correctly. In Question

(3), all the approaches answered the question wrong, including our approach. Even though our approach was able to retrieve at least one correct explanation, it was not able to combine the information to answer and was distracted by lexical noise. These shortcomings indicate that more work can be done, and different constraints can be experimented with for combining facts.

Question (1): Fanning can make a wood fire burn hotter because the fanning: **Correct Answer:** adds more oxygen needed for burning. PathNet **Answer**: provides the energy needed to keep the fire going. **Explanations**: *(i)* fanning a fire increases the oxygen near the fire, *(ii)* placing a heavy blanket over a fire can be used to keep oxygen from reaching a fire ExplanationLP **Answer**: increases the amount of wood there is to burn. **Explanations**: *(i)* more burning causes fire to be hotter, *(ii)* wood burns *Diff*-Explainer ExplanationLP **Answer**: adds more oxygen needed for burning. **Explanations**: *(i)* more burning causes fire to be hotter, *(ii)* fanning a fire increases the oxygen near the fire
Question (2): Which type of graph would best display the changes in temperature over a 24 hour period? **Correct Answer:** line graph. PathNet **Answer**: circle/pie graph. **Explanations**: *(i)* a line graph is used for showing change ; data over time ExplanationLP **Answer**: circle/pie graph. **Explanations**: *(i)* 1 day is equal to 24 hours, *(ii)* a circle graph; pie graph can be used to display percents; ratios *Diff*-Explainer ExplanationLP **Answer**: line graph. **Explanations**: *(i)* a line graph is used for showing change; data over time, *(ii)* 1 day is equal to 24 hours
Question (3): Why has only one-half of the Moon ever been observed from Earth? **Correct Answer:** The Moon rotates at the same rate that it revolves around Earth.. PathNet **Answer**: The Moon has phases that coincide with its rate of rotation. **Explanations**: *(i)* the moon revolving around ; orbiting the Earth causes the phases of the moon, *(ii)* a new moon occurs 14 days after a full moon ExplanationLP **Answer**: The Moon does not rotate on its axis. **Explanations**: *(i)* the moon rotates on its axis, *(ii)* the dark half of the moon is not visible *Diff*-Explainer ExplanationLP **Answer**: The Moon is not visible during the day. **Explanations**: *(i)* the dark half of the moon is not visible, *(ii)* a complete revolution; orbit of the moon around the Earth takes 1; one month

Table 4.6: Example of predicted answers and explanations (Only *CENTRAL* explanations) obtained from our model with different levels of fine-tuning.

4.5 Conclusion

Research Objective 3: *Build a hybrid framework for multi-hop inference that combines constraint-based optimization layers with pre-trained neural representations, enabling end-to-end differentiability for explanation-based inference with optimization-based solvers*

We presented a novel framework for encoding explicit and controllable assumptions as an end-to-end learning framework for question answering. We empirically demonstrated how incorporating these constraints in broader Transformer-based architectures can improve answer and explanation selection. The presented framework adopts constraints from TupleILP and ExplanationLP, but *Diff*-Explainer can be extended to encode different constraints with varying degrees of complexity.

Diff-Explainer is the first work investigating the end-to-end integration of constraint-based solvers and latent neural representations for explanatory inference to the best of our knowledge.

- **RQ3.1**: *Do incorporating constraint solvers with transformers improve performance when compared to the non-differentiable solver?*

  In Section 4.4.1 we demonstrated that *Diff*-Explainer with ExplanationLP and TupleILP outperforms the respective non-differentiable solvers by 13.3% and 8.91% for answer selection for WorldTree corpus and 10.12% and 2.74% for ARC corpus. Additionally, in Section 4.4.2, we also showed improved performance for explanation selection performance over the non-differentiable solvers. In summary, we can conclude that *incorporating constraint solvers with transformers via Differentiable Convex Optimization Solvers approximated by Semi-definite relaxations improve performance compared to the non-differentiable solver*.

- **RQ3.2**: *Does incorporating constraint solvers with transformers demonstrate better robustness in inference to increasing distracting noise compared to transformer-based models?*

  As shown in Section 4.4.3, there is a sudden drop for BERT_{Large} and BERT_{Base} with $k = 10$. In contrast, our model's performance increases as we add distracting information. These results indicate our model's robustness to distracting knowledge. In addition, *Diff*-Explainer has demonstrated improved performance in answer selection (See Table 4.2). The best *Diff*-Explainer setting (with ExplanationLP constraints and fine-tuned on both answer and explanations) outperforms

the best transformer-based models with and without explanations by 12.16% and 21.85%. In summary, we can conclude that *incorporating constraint solvers with transformers via Differentiable Convex Optimization Solvers approximated by Semi-definite relaxations leads to robust reasoning when compared to transformer-based only models*.

4.6 Scope and Limitations

We noticed that fine-tuning the approach with explanations were not yielding the same level of performance increase in explanation selection as it did for answer selection. We hypothesize that the semi-definite approximation of the ILP formulation is leading to sub-optimal predictions for node selection.

Similar to ExplanationLP, *Diff*-Explainer is limited to multiple-choice question answering. If we were to adopt this approach to span-selection, one possible way is to convert span-selection questions into multiple-choice by extracting potential answer spans from the text (Du & Cardie, 2018).

Moreover, the best performing configuration of *Diff*-Explainer results from integrating ExplanationLP constraints. As noted in Section 3.6, ExplanationLP relies on the existence of a corpus of core scientific statements (abstract facts). In our case, we were aided by the existence of WorldTree corpus (Jansen et al., 2018).

4.7 Reproducibility

The rest of the section details the hyper parameters, code bases and datasets used in our approach to reproduce our experiments.

4.7.1 *Diff*-Explainer

External code-bases

- Differentiable convex optimization code-base:
  https://locuslab.github.io/2019-10-28-cvxpylayers/
- Sentence Transformers code-base:
  https://huggingface.co/sentence-transformers/all-mpnet-base-v2

Hyperparameters

We experiment with a range of hyperparameters and pick the hyperparameters with the best performance. The hyperparameter value range tested:

- learning rate: $\{1e-4,\ 5e-5,\ 1e-6\}$
- warmup steps : $\{0,5,\ 10,\ 20\}$
- weight decay: $\{0.0,\ 1e-3,\ 1e-6\}$

Hyperparameter values chosen are as follows:

- gradient accumulation steps: 1
- learning rate: 1e-5
- weight decay: 0.0
- adam epsilon: 1e-8
- warmup steps: 0
- max grad norm: 1.0
- seed: 42

We fine-tuned using 4 Tesla V100 GPUs for 10 epochs in total with batch size 32.

Code

The code for reproducing the *Diff*-Explainer and the experiments described in this chapter are attached with the code appendix and will be available at the following GitHub repository: https://anonymous-url.com.

4.7.2 Approx-TupleILP

We employed Bayesian optimization with the Gaussian process to find the optimal values for $w_1,\ w_2,\ w_3$. We used the https://github.com/fmfn/BayesianOptimization: Bayesian-Optimization python library to implement the code. These parameters are as follows:

- Gaussian Kernels:
  - https://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.kernels.RationalQuadratic.html: RationalQuadratic Kernel with default parameters

  - https://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.kernels.WhiteKernel.html: WhiteKernel with noise level of 1e-5, noise level bounds (1e-10, 1e1) and rest of the default parameters

- Number of iterations: 30
- alpha (α): 1e-8
- random state: 42

4.7.3 Empirical Evaluation

The BERT model was taken from the Hugginface Transformers(https://github.com/huggingface/transformers) library and fine-tuned using 4 Tesla V100 GPUs for 10 epochs in total with batch size 16 for BERT_{Large} and 32 for BERT_{Base}.

The hyperparameters adopted for BERT are as follows:

- gradient accumulation steps: 1
- learning rate: 1e-5
- weight decay: 0.0
- adam epsilon: 1e-8
- warmup steps: 0
- max grad norm: 1.0
- seed: 42

4.7.4 Data

WorldTree Dataset: Data can be obtained from: http://cognitiveai.org/explanationbank/
ARC-Challenge Dataset: https://allenai.org/data/arc. Only used the Challenge split.

Chapter 5

Diff-Comb Explainer: Differentiable Blackbox Combinatorial Solvers for Explanation-based Multi-hop Inference

This chapter is based on the paper "Going Beyond Approximation: Encoding Constraints for Explainable Multi-hop Inference via Differentiable Combinatorial Solvers". The current version of the paper can be found in https://arxiv.org/abs/2208.03339.

5.1 Introduction

In an attempt to combine the best of both worlds, in Chapter 4 proposed a novel neuro-symbolic framework (*Diff*-Explainer) that integrates explicit constraints with neural representations via Differentiable Convex Optimization Layers (Agrawal, Amos, et al., 2019). *Diff*-Explainer combines constraint optimization solvers with Transformers-based representations, enabling end-to-end training for explanation-based multi-hop inference. The *non-differenitability* of ILP solvers is alleviated by approximating the constraints using Semi-Definite programming (Helmberg, 2000). This approximation usually requires non-trivial transformations of ILP formulations into convex optimization problems.

Since constraint-based multi-hop inference is typically framed as an optimal subgraph selection problem achieved via a binary optimization $(0, 1)$, The semi-definite relaxation employed in *Diff*-Explainer necessitates a continuous relaxation of the discrete variables (from $\{0, 1\}$ to $[0, 1]$). While this process can provide tight approximations for ILP problems, this relaxation can still lead to sub-optimal results in practice (Thapper

& Živný, 2017; Yoshida, 2011) leading to erroneous answer and explanation prediction.

To improve on these limitations, we propose *Diff*-Comb Explainer, a novel neuro-symbolic architecture based on *Differentiable BlackBox Combinatorial solvers* (Pogančić et al., 2019). The proposed algorithm transforms a combinatorial optimization solver into a composable building block of a neural network. Differentiable BlackBox Combinatorial solvers (DBCS) achieves this by leveraging the minimization structure of the combinatorial optimization problem, computing a gradient of continuous interpolation to address the *non-differenitability* of ILP solvers. In contrast to *Diff*-Explainer, DBCS makes it possible to compute exact solutions for the original ILP problem under consideration, approximating the gradient.

Our experiments on multi-hop question answering with constraints adopted from ExplanationLP yielded an improvement of 11% over non-differentiable solvers and 2.08% over *Diff*-Explainer. Our approach also produces accurate inference chains with *Diff*-Comb Explainer outperforming the non-differentiable solver and *Diff*-Explainer by 8.41% and 3.63% for explanation selection, respectively.

In summary, the contributions of the chapter are as follows:

1. A novel constrained-based natural language solver combines the differentiable black box combinatorial solver with transformer-based architectures.

2. Empirically demonstrates that differentiable combinatorial solvers combined with transformer architectures provide improved performance for explanation and answer selection compared to the differentiable and non-differentiable counterparts.

3. Demonstrate that *Diff*-Comb Explainer better reflects the underlying inference process for the answer prediction compared to differentiable and non-differentiable counterparts.

5.2 Differentiable Blackbox Combinatorial Optimization Solver

Given the following bounded integer problem:

$$\min_{x \in X} c \cdot x \qquad \text{subject to} \qquad Ax \leq b, \tag{5.1}$$

where $X \in \mathbb{Z}^n$, $c \in \mathbb{R}^n$, x are the variables, $A = [a_1, \ldots, a_m] \in \mathbb{R}^{m \times n}$ is the matrix of

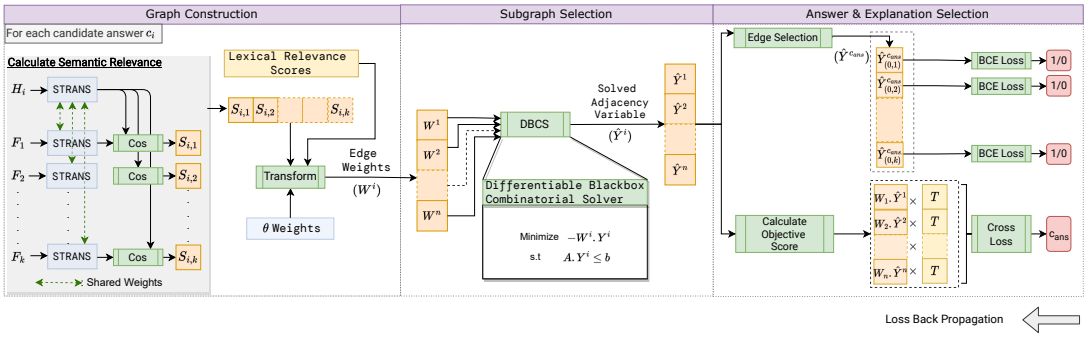

Figure 5.1: End-to-end architectural diagram of *Diff-Comb Explainer*. The integration of Differentiable Blackbox Combinatorial solvers will result in better explanation and answer prediction.

constraint coefficients and $b \in \mathbb{R}^m$ is the bias term. The output of the solver $g(c)$ returns the $\arg\min_{x \in X}$ of the integer problem.

Differentiable Combinatorial Optimization Solver (Pogančić et al., 2019) (DBCS) assumes that A, b are constant and the task is to find the $\mathrm{d}L/\mathrm{d}c$ given global loss function L with respect to solver output x at a given point $\hat{x} = g(\hat{c})$. However, a small change in c is *typically* not going to change the optimal ILP solution resulting in the true gradient being zero.

In order to solve this problem, the approach simplifies by considering the linearisation f of L at the point $\hat{x}$.

$$f(x) = L(\hat{x}) + \frac{\mathrm{d}L}{\mathrm{d}x}(\hat{x}) \cdot (x - \hat{x}) \tag{5.2}$$

to derive:

$$\frac{\mathrm{d}f(g(c))}{\mathrm{d}c} = \frac{\mathrm{d}L}{\mathrm{d}c} \tag{5.3}$$

By introducing the linearisation, the focus is now on differentiating the piecewise constant function $f(g(c))$. The approach constructs a continuous interpolation of $f(g(c))$ by function $f_\lambda(w)$. Here the hyper-parameter $\lambda > 0$ controls the trade-off between *informativeness of the gradient* and *faithfulness to the original function*.

5.3 *Diff*-Comb Explainer: Differentiable Blackbox Combinatorial Solver for Explanation-based Multi-Hop Inference

As illustrated in Figure 5.1, *Diff*-Comb Explainer has 3 major parts: *Graph Construction*, *Subgraph Selection* and *Answer/Explanation Selection*. In *Graph Construction*, for each candidate answer c_i we construct graph $G^i = (V^i,\ E^i,\ W^i)$ where the $V^i = \{h_i\} \cup \{F\}$ and weights W^i_{ik} of each edge E^i_{ik} denote how relevant a fact f_k is with respect to the hypothesis h_i. These edge weights (W^i_{ik}) are calculated using a weighted (θ) sum of scores calculated using transformer-based (*STrans*) embeddings and lexical overlap.

In the *Subgraph Selection* step, for each G^i Differentiable Blackbox Combinatorial Solver (DBCS) with constraints are applied to extract subgraph $\tilde{G}$. In this approach, we adopt the constraints proposed for ExplanationLP.

Finally, in *Answer/Explanation Selection* the model is to predict the correct answer c_{ans} and relevant explanations F_{exp}. During training time, the loss is calculated based on gold answer/explanations to fine-tune the transformers (*STrans*) and weights (θ).

The rest of the section explains each of the components in detail.

5.3.1 Graph Construction

In order to facilitate grounding abstract chains, the retrieved facts F are classified into *grounding* facts $F_G = \{f^g_1, f^g_2, f^g_3, ..., f^g_l\}$ and abstract facts $F_A = \{f^a_1, f^a_2, f^a_3, ..., f^a_m\}$ such that $F = F_A \cup F_G$ and $l + m = k$.

Similarly to *Diff*-Explainer, we use two relevance scores: semantic and lexical scores, to calculate the edge weights. We use a Sentence-Transformer (STrans) (Reimers et al., 2019) bi-encoder architecture to calculate the semantic relevance. The semantic relevance score from STrans is complemented with the lexical relevance score. The semantic and lexical relevance scores are calculated as follows:

Semantic Relevance (s): Given a hypothesis h_i and fact f_j we compute sentence vectors of $\vec{h_i} = STrans(h_i)$ and $\vec{f_j} = STrans(f_j)$ and calculate the semantic relevance score using cosine-similarity as follows:

$$s_{ij} = S(\vec{h_i},\ \vec{f_j}\ ) = \frac{\vec{h_i} \cdot \vec{f_j}}{\|\vec{h_i}\|\|\vec{f_j}\|} \tag{5.4}$$

Lexical Relevance (l): The lexical relevance score of hypothesis h_i and f_j is given by

the percentage of overlaps between unique terms (here, the function trm extracts the lemmatized set of unique terms from the given text):

$$l_{ij} = L(h_i,\ f_j) = \frac{|trm(h_i) \cap trm(f_j)|}{max(|trm(h_i)|, |trm(f_j)|)} \tag{5.5}$$

Given the above scoring function, we construct the edge weights matrix (W) as follows:

$$W^i_{jk} = \begin{cases} -\theta_{gg} l_{jk} & (j,k) \in F_G \\ -\theta_{aa} l_{jk} & (j,k) \in F_A \\ \theta_{ga} l_{jk} & j \in F_G, k \in F_A \\ \theta_{qgl} l_{jk} + \theta_{qgs} s_{jk} & j \in F_G, k = h_i \\ \theta_{qal} l_{jk} + \theta_{qal} s_{jk} & j \in F_A, k = h_i \end{cases} \tag{5.6}$$

Here relevance scores are weighted by θ parameters which are clamped to $[0,1]$.

5.3.2 Subgraph Selection via Differentiable Blackbox Combinatorial Solvers

Given the above premises, the objective function is defined as:

$$\min \qquad -1(W^i \cdot Y^i) \tag{5.7}$$

We adopt the edge variable $Y^i \in \{0,1\}^{(n+1)\times(n+1)}$ where $Y^i_{j,k}$ $(j \neq k)$ takes the value of 1 iff edge E^i_{jk} belongs to the subgraph and Y^i_{jj} takes the value of 1 iff V^i_j belongs to the subgraph.

Given the above variable, the constraints are defined as follows:

Answer selection constraint The candidate hypothesis should be part of the induced subgraph:

$$\sum_{j \in \{h_i\}} Y^i_{jj} = 1 \tag{5.8}$$

Edge and Node selection constraint If node V^i_j and V^i_k are selected then edges E^i_{jk} and E^i_{kj} will be selected. If node V^i_j is selected, then edge E_{jj} will also be selected:

$$Y^i_{jk} \leq Y^i_{jj} \qquad \forall\,(j,k) \in E \tag{5.9}$$

$$Y^i_{jk} \leq Y_{kk} \qquad \forall\,(j,k) \in E \tag{5.10}$$

$$Y^i_{jk} \geq Y_{jj} + Y_{kk} - 1 \qquad \forall\,(j,k) \in E \tag{5.11}$$

Abstract fact selection constraint Limit the number of abstract facts selected to M:

$$\sum_i Y^i_{jj} \leq M \qquad \forall j \in F_A \tag{5.12}$$

5.3.3 Answer and Explanation Selection

The solved adjacency variable $\hat{Y}^i$ represents the selected edges for each candidate answer choice c_i. Not all datasets provide gold explanations. Moreover, even when the gold explanations are available, they are only available for the correct answer with no explanations for the *wrong* answer.

In order to tackle these shortcomings and ensure end-to-end differentiability, we use the softmax (σ) of the objective score ($W^i \cdot \hat{Y}^i$) as the probability score for each choice.

We multiply each objective score $W^i \cdot \hat{Y}^i$ value by the temperature hyperparameter (T) to obtain soft probability distributions γ^i (where $\gamma^i = (W^i \cdot \hat{Y}^i) \cdot T$). The aim is for the correct answer c_{ans} to have the highest probability.

In order to achieve this aim, we use the cross entropy loss l_c as follows to calculate the answer selection loss $\mathcal{L}_{ans}$ as follows:

$$\mathcal{L}_{ans} = l_c(\sigma(\gamma^1,\, \gamma^2,\, \cdots \gamma^n),\, c_{ans}) \tag{5.13}$$

If gold explanations are available, we complement $\mathcal{L}_{ans}$ with explanation loss $\mathcal{L}_{exp}$. We employ binary cross entropy loss l_b between the selected explanatory facts and gold explanatory facts F_{exp} for the explanatory loss as follows:

$$\mathcal{L}_{exp} = l_b(\hat{Y}^{ans}[f_1,\, f_2,\, \ldots,\, f_k],\, F_{exp}) \tag{5.14}$$

We calculate the total loss ($\mathcal{L}$) as weighted by hyperparameters λ_{ans}, λ_{exp} as follows:

$$\mathcal{L} = \lambda_{ans}\mathcal{L}_{ans} + \lambda_{exp}\mathcal{L}_{exp} \tag{5.15}$$

The pseudo-code to train *Diff*-Comb Explainer end-to-end is summarized in Algorithm 4.

Algorithm 4: Training *Diff*-Comb Explainer.

Data: $A,\ b \leftarrow$ Multi-hop Inference Constraints
Data: $f_w \leftarrow$ Graph weight Function
Data: $\lambda \leftarrow$ Hyperparameter for DBCS interpolation
$epoch \leftarrow 0$;
while $epoch \leq max\_epochs$ **do**
  foreach $h_i \in H$ **do**
    $G^i \leftarrow$ fact-graph-construction$(h_i,\ F)$;
    $l^i \leftarrow L(h_i,\ F)$;
    $\theta \leftarrow clamp([0,\ 1])$;
    $\vec{F} \leftarrow STrans(F)$;
    $\vec{h_i} \leftarrow STrans(h_i)$;
    $s^i \leftarrow S(\vec{h_i},\ \vec{F})$;
    $W^i \leftarrow f_w(s^i,\ l^i;\ \theta)$;
    $\hat{Y}^i \leftarrow DBCS(-W^i,\ A,\ b;\lambda)$;
    $\gamma^i \leftarrow (W \cdot \hat{y}) \cdot T$;
  end
  $\mathcal{L}_{ans} = l_c(\sigma(\gamma^1,\ \gamma^2,\ \cdots\ \gamma^n),\ c_{ans})$;
  if F_{exp} *is available* **then**
    $\mathcal{L}_{exp} = l_b(\hat{Y}^{ans}[f_1,\ f_2,\ \ldots,\ f_k],\ F_{exp})$;
    $\mathcal{L} = \lambda_{ans}\mathcal{L}_{ans} + \lambda_{exp}\mathcal{L}_{exp}$;
  else
    $\mathcal{L} = \mathcal{L}_{ans}$;
  end
  update θ, *STrans* using AdamW optimizer by minimizing *loss*;
  $epoch \leftarrow epoch + 1$;
end
Result: Store best θ and *STrans*

5.4 Empirical Evaluation

5.4.1 Answer and Explanation Selection

We use the WorldTree corpus (Xie et al., 2020) for training the evaluation of explanation and answer selection. The 4,400 question and explanations in the WorldTree corpus are split into three different subsets: *train-set*, *dev-set* and *test-set*. We use the *dev-set* to assess the explanation selection performance since the explanations for *test-set* are not publicly available.

Model	**Explanation Selection (*dev*)**						**Answer Selection (*test*)**
	Precision		**Explanatory Consistency**			**Faithfulness**	
	@2	**@1**	**@3**	**@2**	**@1**		
Baselines							
$BERT_{Base}$	-	-	-	-	-	-	45.43
$BERT_{Large}$	-	-	-	-	-	-	49.63
Fact Retrieval (FR) Only	30.19	38.49	21.42	15.69	11.64	-	-
$BERT_{Base}$ + FR	-	-	-	-	-	52.65	58.06
$BERT_{Large}$ + FR	-	-	-	-	-	51.23	59.32
ExplanationLP	40.41	51.99	29.04	14.14	11.79	71.11	62.57
Diff-Explainer	41.91	56.77	39.04	20.64	17.01	72.22	71.48
***Diff*-Comb Explainer**							
- Answer selection only	45.75	61.01	**49.04**	29.99	18.88	73.37	72.04
- Answer and explanation selection	**47.57**	**63.23**	43.33	**33.36**	**20.71**	**74.47**	**73.46**

Table 5.1: Comparison of explanation and answer selection of *Diff*-Comb Explainer against other baselines. Explanation Selection was carried out on the *dev* set as the *test* explanation was not public available.

Baselines: We use the following baselines to compare against our approach for the WorldTree corpus:

1. **$BERT_{Base}$ and $BERT_{Large}$** (Devlin et al., 2019): To use BERT for this task, we concatenate every hypothesis with k retrieved facts, using the separator token `[SEP]`. We use the HuggingFace (Wolf et al., 2019) implementation of *BertForSequenceClassification*, taking the prediction with the highest probability for the positive class as the correct answer.

2. **ExplanationLP**: Non-differentiable version of ExplanationLP. Using the constraints stated in Section 5.3, we fine-tune the θ parameters using Bayesian optimization and frozen STrans representations. This baseline aims to evaluate the impact of end-to-end fine-tuning over the non-differentiable solver.
3. ***Diff*-Explainer**: *Diff*-Explainer has already exhibited better performance over other explanation-based multi-hop inference approaches, including ILP-based approaches including TableILP (Khashabi et al., 2016), TupleILP (Khot et al., 2017) and graph-based neural approach PathNet (Kundu et al., 2019). Similar to our approach, we use ExplanationLP constraints with *Diff*-Explainer. We use similar hyperparameters and knowledge base used in Chapter 4.

Experimental Setup: We follow the similar experimental setup used in *Diff*-Explainer as follows:

- *Sentence Transformer Model*: ALL-MPNET-BASE-V2 (Song et al., 2020).
- *Fact retrieval representation*: ALL-MPNET-BASE-V2 trained with gold explanations of WorldTree Corpus to achieve a Mean Average Precision of 40.11 in the dev-set.
- *Fact retrieval*: FAISS retrieval (Johnson et al., 2019) using pre-cached representations.
- *Background knowledge*: 5000 abstract facts from the WorldTree table store (WTree) and over 100,000 *is-a* grounding facts from ConceptNet (CNet) (Speer et al., 2017).
- The experiments were carried out for $k = \{1,2,3,5,10,20,30,40,50\}$ and the best configuration for each model is selected.
- The hyperparameters $\lambda, \lambda_{ans}, \lambda_{exp}, T$ were fine-tuned for 50 epochs using the Adpative Experimentation Platform.
- M=2 for ExplanationLP, *Diff*-Explainer and *Diff*-Comb Explainer.

Metrics The answer selection is evaluated using accuracy. For evaluation of explanation selection, we use Precision@K. In addition to Precision@K, we introduce two new metrics to evaluate the truthfulness of the answer selection to the underlying inference. The metrics are as follows:

Explanatory Consistency@K: Question/answer pair with similar explanations indicates similar underlying inference (Atanasova et al., 2022). The expectation is that similar underlying inference would produce similar explanations (Valentino et al., 2021). Given a test question Q_t and retrieved explanations E_t we find set of Questions $Q_t^s = \{Q_t^1,\ Q_t^2,\ \ldots\}$ with at least K overlap gold explanations along with the retrieved explanations $E_t^s = \{e_t^1,\ e_t^2,\ \ldots\}$. Given this premise, Explanatory Consistency@K is

defined as follows:

$$\frac{\sum_{e_t^i \in E_t^s} [e_t^i \in E_t]}{\sum_{e_t^i \in E_t^s} |e_t^i|} \tag{5.16}$$

Explanatory Consistency measures out of questions/answer pairs with at least K similar gold explanations and how many of them share a common retrieved explanation.

Faithfulness: The aim is to measure how much percentage of the correct prediction is derived from correct inference and wrong prediction is derived from wrong inference over the entire set. Let's say that the set of questions correctly answered as A_{Q_c}, wrongly answered questions A_{Q_w}, set of questions with at least one correctly retrieved explanation as A_{Q_1} and set of questions where no correctly retrieved explanations A_{Q_0}. Given this premise, Faithfulness is defined as follows:

$$\frac{|A_{Q_w} \cap A_{Q_0}| + |A_{Q_c} \cap A_{Q_1}|}{|A_{Q_c} \cup A_{Q_w}|} \tag{5.17}$$

A higher faithfulness implies that the underlying inference process is reflected in the final answer prediction.

Table 5.1 illustrates the explanation and answer selection performance of *Diff*-Comb Explainer and the baselines. We report scores for *Diff*-Comb Explainer trained for only the answer and optimized jointly for answer and explanation selection.

Since BERT does not provide explanations, we use facts retrieved from the fact retrieval for the best k configuration ($k = 3$) as explanations. We also report the scores for BERT without explanations.

We draw the following conclusions from the results obtained in Table 5.1 (The performance increase here are expressed in absolute terms):

1. *Diff*-Comb Explainer improves answer selection performance over the non-differentiable solver by 9.47% with optimizing only on answer selection and 10.89% with optimizing on answer and explanation selection. This observation underlines the impact of the end-to-end fine-tuning framework. We can also observe that strong supervision with optimizing explanation selection yields better performance than weak supervision with answer selection.
2. *Diff*-Comb Explainer outperforms the best transformer-based model by 14.14% for answer selection. This increase in performance demonstrates that integrating constraints with transformer-based architectures leads to better performance.
3. *Diff*-Comb Explainer outperform the best *Diff*-Explainer configuration (answer and explanation selection) by 0.56% even in the weak supervision setting (answer

only optimization). *Diff*-Comb Explainer also outperform *Diff*-Explainer by 1.98% in the best setting.

4. *Diff*-Comb Explainer is better for selecting relevant explanations over the other constraint-based solvers. *Diff*-Comb Explainer outperforms the non-differentiable solver at Precision@K by 8.41% (k =1) and 6.05% (k =2). *Diff*-Comb Explainer also outperforms *Diff*-Explainer by 3.63% (k =1) and 4.55% (k =2). The improvement of Precision@K over the Fact Retrieval only (demonstrated with BERT + FR) by 16.98% (k =1) and 24.74% (k =2) underlines the robustness of our approach to noise propagated by the upstream fact retrieval.
5. *Diff*-Comb Explainer also exhibits higher Explanatory Consistency over the other solvers. This performance shows that the optimization model is learning and applying consistent inference across different instances.
6. Answer prediction by *black-box* models like BERT do not reflect the explanation provided. This fact is indicated by the low Faithfulness score obtained by both BERT_{Base}/BERT_{Large}. In contrast, the high constraint-based solver's Faithfulness scores emphasize how the underlying inference reflects on the final prediction. In particular, *Diff*-Explainer and *Diff*-Comb Explainer approach performs better than the non-differentiable model.

In summary, even though *Diff*-Explainer and *Diff*-Comb Explainer approaches use the same set of constraints, *Diff*-Comb Explainer model yields better performance, indicating that ILP solvers generate those accurate predictions are better than approximated sub-optimal results.

5.4.2 Knowledge aggregation with increasing distractors

One of the key characteristics identified by *Diff*-Explainer is the robustness of *Diff*-Explainer to distracting noise. In order to evaluate if *Diff*-Comb Explainer also exhibits the same characteristics, we ran *Diff*-Comb Explainer for the increasing number of retrieved facts k and plotted the answer selection accuracy for WorldTree in Figure 5.2.

As illustrated in the Figure, similar to *Diff*-Explainer, our approach performance remains stable with increasing distractors. We also continue to outperform *Diff*-Explainer across all sets of k.

BERT performance drops drastically with increasing distractors. This phenomenon is in line with existing work (Yadav et al., 2019b). We hypothesize that with increasing distractors, BERT overfits quickly with spurious inference correlation. On the other hand, our approach circumvents this problem with the inductive bias provided by the

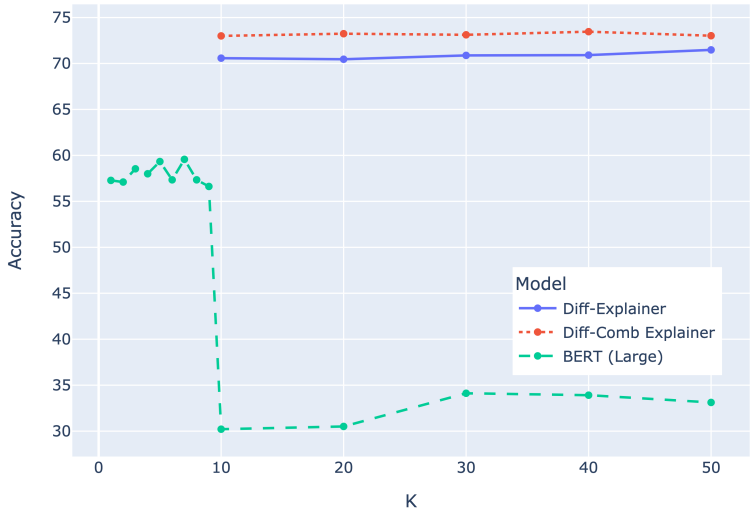

Figure 5.2: Comparison of accuracy for different number of retrieved facts.

constraint optimization layer.

5.4.3 Comparing Answer Selection with ARC Baselines

Table 5.2 presents a comparison of publicly reported baselines on the ARC Challenge-Corpus (Clark et al., 2018) and our approach. These questions have proven to be challenging to answer for other LP-based question answering and neural approaches.

While models such as UnifiedQA (Khashabi et al., 2020) and AristoBERT (Xu et al., 2021) have demonstrated performance of 81.14 and 68.95, they have been trained on other question-answering datasets, including RACE (Lai et al., 2017). Moreover, despite its performance, UnifiedQA does not provide explanations supporting its inference.

In Table 5.2, to provide a rigorous comparison, we only list models that have been trained *only* on the ARC corpus and provide explanations supporting its inference to ensure a fair comparison. Here the explainability column indicates if the model delivers an explanation for the predicted answer. A subset of the models produces evidence for the answer but remains intrinsically black-box. These models have been marked as *Partial*.

As illustrated in the Table 5.2, *Diff*-Comb Explainer outperforms the best non-differentiable constraint-solver model (ExplanationLP) by 2.8%. We also outperform a transformer-only model AutoRocc by 1.97%. While *Diff*-Comb Explainer improvement

Model	Explainable	Accuracy
BERT$_{Large}$	No	35.11
IR Solver (Clark et al., 2016)	Yes	20.26
TupleILP (Khot et al., 2017)	Yes	23.83
TableILP (Khashabi et al., 2016)	Yes	26.97
ExplanationLP	Yes	40.21
DGEM (Clark et al., 2016)	Partial	27.11
KG^2 (Y. Zhang et al., 2018)	Partial	31.70
ET-RR (Ni et al., 2019)	Partial	36.61
Unsupervised AHE (Yadav et al., 2019a)	Partial	33.87
Supervised AHE (Yadav et al., 2019a)	Partial	34.47
AutoRocc (Yadav et al., 2019b)	Partial	41.24
Diff-Explainer (ExplanationLP)	Yes	**42.95**
Diff-Comb Explainer (ExplanationLP)	Yes	**43.21**

Table 5.2: ARC challenge scores compared with other Fully or Partially explainable approaches trained *only* on the ARC dataset.

over *Diff*-Explainer is small, we still demonstrate performance improvements for answer selection. On top of performances obtained for explanation and answer selection with WorldTree corpus, we have also established better performances than leaderboard approaches.

5.4.4 Qualitative Analysis

Table 5.3 illustrates some of the answers and explanations extracted for ExplanationLP, *Diff*-Explainer and *Diff*-Comb Explainer. Both explanations and answer predictions in Question (1) are entirely correct for *Diff*-Comb Explainer. In this Example, both ExplanationLP and *Diff*-Explainer have failed to retrieve any correct explanations or predict the correct answer. Both the approaches are distracted by the strong lexical overlaps with the wrong answer.

Question (2) at least one explanation is correct and a correct answer prediction for *Diff*-Comb Explainer. In the Example provided, *Diff*-Explainer provides the correct answer prediction with both the retrieved facts not being explanatory. *Diff*-Explainer arrives at the correct answer prediction with no explanation addressing the correct answer.

In Question (3), both *Diff*-Comb Explainer and *Diff*-Explainer provide the correct answer but with both facts not being explanations. The aforementioned qualitative (Question 1 and 2) and quantitative measures (Explanatory Consistency@K, Faithfulness) indicate how the underlying explanatory inference results in the correct prediction; there are cases where false inference still leads to the correct answer with *Diff*-Comb Explainer as well. In this case, the inference is distracted by the strong lexical overlaps irrelevant to the question.

However, from the qualitative analysis, we can conclude that the explanation-based inference that happens with *Diff*-Comb Explainer is more robust and coherent when compared to the *Diff*-Explainer and non-differentiable models.

Question (1): Which measurement is best expressed in light-years?: **Correct Answer:** the distance between stars in the Milky Way. ExplanationLP **Answer**: the time it takes for planets to complete their orbits. **Explanations**: *(i)* a complete revolution; orbit of a planet around its star takes 1; one planetary year, *(ii)* a light-year is used for describing long distances *Diff*-Explainer **Answer**: the time it takes for planets to complete their orbits. **Explanations**: *(i)* a light-year is used for describing long distances, *(ii)* light year is a measure of the distance light travels in one year *Diff*-Comb Explainer **Answer**: the distance between stars in the Milky Way. **Explanations**: *(i)* light years are a astronomy unit used for measuring length, *(ii)* stars are located light years apart from each other
Question (2): Which type of precipitation consists of frozen rain drops?: **Correct Answer:** sleet. ExplanationLP **Answer**: snow. **Explanations**: *(i)* precipitation is when snow fall from clouds to the Earth, *(ii)* snow falls *Diff*-Explainer **Answer**: sleet. **Explanations**: *(i)* snow falls, *(ii)* precipitation is when water falls from the sky *Diff*-Comb Explainer **Answer**: sleet. **Explanations**: *(i)* sleet is when raindrops freeze as they fall, *(ii)* sleet is made of ice
Question (3): Most of the mass of the atom consists of?: **Correct Answer:** protons and neutrons. ExplanationLP **Answer**: neutrons and electrons. **Explanations**: *(i)* neutrons have more mass than an electron, *(ii)* neutrons have more mass than an electron *Diff*-Explainer **Answer**: protons and neutrons. **Explanations**: *(i)* the atomic mass is made of the number of protons and neutrons, *(ii)* precipitation is when water falls from the sky *Diff*-Comb Explainer **Answer**: protons and neutrons. **Explanations**: *(i)* the atomic mass is made of the number of protons and neutrons, *(ii)* precipitation is when water falls from the sky

Table 5.3: Example of predicted answers and explanations (Only *CENTRAL* explanations) obtained from *Diff*-Comb Explainer with different levels of fine-tuning.

5.5 Conclusion

Research Objective 3: *Build a hybrid framework for multi-hop inference that combines constraint-based optimization layers with pre-trained neural representations, enabling end-to-end differentiability for explanation-based inference with optimization-based solvers*

This Chapter proposed a novel framework for encoding explicit and controllable assumptions as part of an end-to-end learning framework for explanation-based multi-hop inference using Differentiable Blackbox Combinatorial Solvers (Pogančić et al., 2019). We empirically demonstrated improved answer and explanation selection performance compared with the *Diff*-Explainer. We also demonstrated performance gain and increased robustness to noise when combining constraints with transformer-based architectures. In this Chapter, we adopted the constraints of ExplanationLP, but it is possible to encode more complex inference constraints within the model.

Diff-Comb Explainer builds on the previous Chapter and investigates the combination of symbolic knowledge (expressed via constraints) with neural representations.

- **RQ3.1**: *Do incorporating constraint solvers with transformers improve performance when compared to the non-differentiable solver?*

  In Section 5.4.1, we demonstrated *Diff*-Comb Explainer improves answer selection performance over the non-differentiable solver by 9.47% with optimizing only on answer selection and 10.89% with optimizing on answer and explanation selection. In addition, *Diff*-Comb Explainer outperforms the non-differentiable solver at Precision@K by 8.41% (k =1) and 6.05% (k =2). Our model also exhibits higher Explanatory Consistency (by 20%, 19.22%, 8.92% for $k = 3, 2, 1$ respectively) and Faithfulness (by 3.36%) over non-differentiable solver. Based on this evidence, we can conclude that *incorporating constraint solvers via Differentiable Blackbox Combinatorial Solver with transformers improves performance compared to the non-differentiable solver*.

- **RQ3.2**: *Does incorporating constraint solvers with transformers demonstrate better robustness in inference to increasing distracting noise compared to transformer-based models?*

  In Section 5.4.1, we showed that *Diff*-Comb Explainer outperforms the best transformer-based model by 14.14% for answer selection. The low faithfulness obtained by BERT_{Base}/BERT_{Large} also showed *black-box* models like BERT do

not reflect the explanation provided. In contrast, with our approach, we got significantly better faithfulness scores (an increase of over 21.82%). In summary, we can conclude that *incorporating constraint solvers via Differentiable Blackbox Combinatorial Solver with transformers leads to robust reasoning compared to transformer-based only models*.

5.6 Scope and Limitations

Similar to ExplanationLP, *Diff*-Comb Explainer is limited to multiple-choice question answering. If we were to adopt this approach to span-selection, one possible way is to convert span-selection questions into multiple-choice by extracting potential answer spans from the text (Du & Cardie, 2018).

The experiments showed us that the approach is highly dependent on the λ parameter; we might need to experiment with approaches with no need for this hyperparameter-dependent gradient smoothing, such as CombOptNet (Paulus et al., 2021). However, adopting this module would require non-trivial changes in the architecture.

Moreover, the best performing configuration of *Diff*-Comb Explainer results from integrating ExplanationLP constraints. As noted in Section 3.6, ExplanationLP relies on the existence of a corpus of core scientific statements (abstract facts). In our case, we were aided by the existence of WorldTree corpus (Jansen et al., 2018).

5.7 Reproducibility

The rest of the section details the hyperparameters, code bases and datasets used in our approach to reproduce our experiments.

5.7.1 External code-bases

- Differentiable Blackbox Combinatorial Solvers Examples: https://github.com/martius-lab/blackbox-differentiation-combinatorial-solvers
- Sentence Transformer code-base: https://huggingface.co/sentence-transformers/all-mpnet-base-v2

5.7.2 Integer Linear Programming Optimization

The components of the linear programming system is as follows:

- Solver: Gurobi Optimization https://www.gurobi.com/products/gurobi-optimizer/

The hyperparameters used in the ILP constraints:

- Maximum number of abstract facts (M): 2

Infrastructures used:

- CPU Cores: 32
- CPU Model: Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz
- Memory: 128GB

5.7.3 Hyperparameters

For *Diff*-Comb Explainer we had to fine-tune hyperparameters $\lambda, \lambda_{ans}, \lambda_{exp}, T$. We fine-tune for 50 epochs using the Adpative Experimentation Platform with the seed of 42.

The bounds of the hyperparameters are as follows:

- λ: $[100, 300]$
- λ_{exp}: $[0.0, 1.0]$
- λ_{ans}: $[0.0, 1.0]$
- T: $[1e-2, 100]$

The hyperparameters adopted for our approach are as follows:

- λ: 152
- λ_{exp}: 0.72
- λ_{ans}: 0.99
- T: 8.77
- max epochs: 8
- gradient accumulation steps: 1
- learning rate: 1e-5
- weight decay: 0.0
- adam epsilon: 1e-8
- warmup steps: 0
- max grad norm: 1.0
- seed: 42

The hyperparameters adopted for BERT are as follows:

- gradient accumulation steps: 1
- learning rate: 1e-5
- weight decay: 0.0
- adam epsilon: 1e-8
- warmup steps: 0
- max grad norm: 1.0
- seed: 42

We fine-tuned using 4 Tesla V100 GPUs for 10 epochs in total with batch size 32 for *Base* and 16 for *Large*.

5.7.4 Data

WorldTree Dataset: Data can be obtained from: http://cognitiveai.org/explanationbank/
ARC-Challenge Dataset: https://allenai.org/data/arc. Only used the Challenge split.

Chapter 6

Conclusion & Future Work

6.1 Summary and Conclusions

The **central research problem** stated in Chapter 1 is as follows:

Given a question (Q) and candidate answers $C = \{c_1,\ c_2,\ c_3, ...,\ c_n\}$, the aim is to build a differentiable constraint-based optimization model $Diff_{constrained}$ robust to semantic drift that combines constraint-based solvers and transformers to select the correct answer c_{ans} and explanation F_{ans} that supports the answer.

As underlined in the research problem, in this thesis, we set out to integrate constrained-based optimization solvers with deep learning models and address the problem of semantic drift. Our contributions focused on two dimensions: design constraints that alleviate the problem of semantic drift and address the problem of ILP non-differentiability.

Our motivation to address this challenge was derived from the survey carried out in *Chapter 2*. In *Chapter 2*, we presented a survey of benchmarks and models proposed in Machine Reading Comprehension. We first categorized explanations based on their function in the inference into knowledge-based and operational-based (See Section 2.2). We also identified that explanation-based inference is now more evolving towards training in evaluating abstractive reasoning. Based on the analysis we performed in explanation-supporting benchmarks and explanation-based inference models (See Section 2.6), we identified the effectiveness of hybrid approaches that combines latent representation with structural representation. In particular, constraint-based models based on ILP solvers can provide a mechanism to encode explicit and controllable

assumptions about the structure of the inference. However, these approaches were limited due to their susceptibility to semantic drift and inability to be incorporated with broader deep learning frameworks. These two challenges formed the core motivation of the thesis and the central research problem. The contributions of this Chapter are as follows (Restated from Chapter 2):

Contributions:

1. Provide an analysis of existing benchmarks and models for explanation-based inference in Machine Reading Comprehension

2. Identify the emerging research trends and architectural design for an explainable system

3. Highlight set of challenges and opportunities for future work

In order to address the challenge of semantic drift, Chapter 3 presented a model (ExplanationLP) that reduces the number of hops to two and introduces explicit grounding-abstract chains to perform inference. This approach addresses the first part of the central research problem: *build a constraint-based optimization model robust to semantic drift*

Our experiments on Answer and Explanation selection demonstrated that our model outperforms Transformer-based models - BERT (Devlin et al., 2019) and graph-based model- PathNet (Kundu et al., 2019) (See Section 3.3.1). ExplanationLP also demonstrated a lower degradation with an increasing number of explanation sentences (See Section 3.3.4). The contributions of this Chapter are as follows (Restated from Chapter 3):

Contributions:

1. We present a novel approach that performs explanation-based inference via grounding-abstract chains combining Linear Programming with Bayesian optimization for science question answering.

2. We obtain comparable performance compared to transformers, multi-hop approaches and previous Linear Programming models despite having a significantly lower number of parameters.

3. We demonstrate that our model can generate plausible explanations for answer prediction and validate the importance of grounding-abstract chains via ablation analysis.

In Chapter 3, we employed Bayesian Optimization to fine-tune the parameters in ExplanationLP. However, Bayesian Optimization is intractable to fine-tune the large neural model. Constraint-solvers based on Integer Linear Programming are non-differentiable and cannot be directly integrated with transformer networks to build a broader learning framework. In order to alleviate this problem, we came up with two different solutions:

- In Chapter 4, we introduced *Diff*-Explainer that approximated the ILP formulation using Semi-definite programming, casting the problem as a convex optimization problem. Consequently, allowing us to integrate them into a deep learning network using differentiable convex optimization layers.

- In Chapter 5, we introduced *Diff*-Comb Explainer that employed the Differentiable Blackbox Combinatorial Optimization Solver (DBCs). Introduced by Pogančić et al. (2019), DBCs solve the non-differentiability by introducing a gradient approximation method.

This approach addresses the second part of the central research problem: *build a differentiable constraint-based optimization model*.

Diff-Explainer and *Diff*-Comb Explainer demonstrated better answer and explanation selection performance over *non-differentiable* solvers (See Section 4.4.1, Section 5.4.1). We also showed that these models are more robust to distractors (See Section 5.4.2) and better reflect the underlying inference than transformer-only models (See Section 5.4.1).

With *Diff*-Explainer, We also noticed that fine-tuning the approach with explanations was not yielding the same level of performance increase in explanation selection as it did for answer selection. We hypothesize that the semi-definite approximation of the ILP formulation is leading to sub-optimal predictions for node selection.

We tried to alleviate this problem by incorporating Differentiable Blackbox Combinatorial Solvers for *Diff*-Comb Explainer. Compared to *Diff* -Explainer, we achieved better answer and explanation selection performance with *Diff*-Comb Explainer.

The contributions of these Chapters are as follows (Restated from Chapter 4 and Chapter 5):

Contributions:

1. A novel differentiable framework for multi-hop inference that incorporates constraints via convex optimization layers into broader Transformer-based architectures for *Diff*-Explainer and via black box combinatorial solver with transformer-based architectures for *Diff*-Comb Explainer

2. An extensive empirical evaluation demonstrates that the proposed frameworks allow end-to-end differentiability on downstream QA tasks for both explanation and answer selection, leading to a substantial improvement compared to non-differentiable constraint-based and transformer-based approaches.

3. We demonstrate that *Diff*-Explainer and *Diff*-Comb Explainer are more robust to distracting information in addressing multi-hop inference when compared to Transformer-based models. Additionally, both models better reflect the underlying inference process for the answer prediction compared to Transformer-based models.

In summary, this thesis presented differentiable constraint-based optimization models *Diff*-Explainer and *Diff*-Comb Explainer robust to semantic drift (with ExplanationLP constraints) that combines constraint-based solvers and transformers to select the correct answer c_{ans} and explanation F_{ans} that supports the answer.

6.2 Opportunities for Future Research

Differentiable constraint-based models to other facets of Natural Language Processing As underlined in the conclusion of Chapter 4, constraint-based solvers have been used not only for natural language inference but also relation extraction (L. Chen et al., 2014; Y. Choi et al., 2006; Roth & Yih, 2004), semantic role labeling (Koomen et al., 2005; Punyakanok et al., 2004), sentiment analysis (Y. Choi & Cardie, 2009) and explanation regeneration (A. Gupta & Srinivasaraghavan, 2020). We could adopt the constraints from these approaches to integrate with differentiable constraint layers leading to explainable and robust models across various spectrums of tasks in Natural Language Processing.

Moving beyond multiple-choice question answering Current formulations of constraint-based solvers and the ones we have introduced are also limited to multiple-choice question answering. While the underlying reasoning mechanisms remain universal across all question-answering tasks, moving towards span selection (Z. Yang et al., 2018) or text generation (Reddy et al., 2019) requires a non-trivial adoption of how the framework is designed. The integration could be achieved by following similar ideas used by graph-based (Ding et al., 2019; L. Qiu et al., 2019) or explicit-path-based inference (Dhingra et al., 2020; Nie et al., 2019) approaches.

Experimenting with abstraction and compositionality constraints This work has shown that explicit abstraction can lead to better performance and alleviate the problems arising from semantic drift. Another facet of artificial general intelligence we require is the need for compositionality (Fodor, 1975; Schneider, 2011). Constraint-based solvers have expressive power where it would be possible to define compositional inference as constraints. If abstraction and compositionality could be achieved, this would lead to more robust and generalisable performance when compared to a purely connectionist approach.

Control and debugging deep learning models In our work, we had shown that the control obtained by better domain-specific priors leads to better overall performance. However, our research has not explicitly evaluated the control obtained via the solvers and how control can be improved. If we can establish a method to impose control into deep learning networks for natural language inference via the solvers, this can potentially lead to debugging the black box models.

Bibliography

Agrawal, A., Barratt, S., Boyd, S., Busseti, E., & Moursi, W. (2019). Differentiating through a cone program. *Journal of Applied and Numerical Optimization*, *1*(2), 107–115.

Agrawal, A., Amos, B., Barratt, S., Boyd, S., Diamond, S., & Kolter, J. Z. (2019). Differentiable convex optimization layers. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems*. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2019/file/9ce3c52fc54362e22053399d3181c638-Paper.pdf

Amos, B. (2019). *Differentiable optimization-based modeling for machine learning* (Doctoral dissertation). PhD thesis, Carnegie Mellon University.

Amos, B., & Kolter, J. Z. (2017). Optnet: Differentiable optimization as a layer in neural networks. *International Conference on Machine Learning*, 136–145.

Andreas, J., Rohrbach, M., Darrell, T., & Klein, D. (2016a). Learning to compose neural networks for question answering. *arXiv preprint arXiv:1601.01705*.

Andreas, J., Rohrbach, M., Darrell, T., & Klein, D. (2016b). Neural module networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Asai, A., Hashimoto, K., Hajishirzi, H., Socher, R., & Xiong, C. (2019). Learning to retrieve reasoning paths over wikipedia graph for question answering. *arXiv preprint arXiv:1911.10470*.

Atanasova, P., Simonsen, J. G., Lioma, C., & Augenstein, I. (2022). Diagnostics-guided explanation generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, *36*(10), 10445–10453.

Banerjee, P. (2019). Asu at textgraphs 2019 shared task: Explanation regeneration using language models and iterative re-ranking. *arXiv preprint arXiv:1909.08863*.

Banerjee, P., & Baral, C. (2020). Knowledge fusion and semantic knowledge ranking for open domain question answering. *arXiv preprint arXiv:2004.03101*.

Banerjee, P., Pal, K. K., Mitra, A., & Baral, C. (2019). Careful selection of knowledge to solve open book question answering. *arXiv preprint arXiv:1907.10738*.

Baradaran, R., Ghiasi, R., & Amirkhani, H. (2020). A survey on machine reading comprehension systems. *arXiv preprint arXiv:2001.01582*.

Baral, C., Banerjee, P., Pal, K. K., & Mitra, A. (2020). Natural language qa approaches using reasoning with external knowledge. *arXiv preprint arXiv:2003.03446*.

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., et al. (2018). Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*.

Bauer, L., Wang, Y., & Bansal, M. (2018). Commonsense for generative multi-hop question answering tasks. *arXiv preprint arXiv:1809.06309*.

Bhagavatula, C., Le Bras, R., Malaviya, C., Sakaguchi, K., Holtzman, A., Rashkin, H., Downey, D., Yih, W.-t., & Choi, Y. (2020). Abductive commonsense reasoning. *ICLR*.

Bhakthavatsalam, S., Anastasiades, C., & Clark, P. (2020). Genericskb: A knowledge base of generic statements. *arXiv preprint arXiv:2005.00660*.

Biran, O., & Cotton, C. (2017). Explanation and justification in machine learning: A survey. *IJCAI-17 workshop on explainable AI (XAI)*, *8*.

Boratko, M., Padigela, H., Mikkilineni, D., Yuvraj, P., Das, R., McCallum, A., Chang, M., Fokoue-Nkoutche, A., Kapanipathi, P., Mattei, N., et al. (2018). A systematic classification of knowledge, reasoning, and context within the arc dataset. *Proceedings of the Workshop on Machine Reading for Question Answering*, 60–70.

Bowman, S., Angeli, G., Potts, C., & Manning, C. D. (2015). A large annotated corpus for learning natural language inference. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 632–642.

Brochu, E., Cora, V. M., & De Freitas, N. (2010). A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, *33*, 1877–1901.

Busseti, E., Moursi, W. M., & Boyd, S. (2019). Solution refinement at regular points of conic problems. *Computational Optimization and Applications*, *74*(3), 627–643.

Camburu, O.-M., Rocktäschel, T., Lukasiewicz, T., & Blunsom, P. (2018). E-snli: Natural language inference with natural language explanations. *Advances in Neural Information Processing Systems*, 9539–9549.

Chen, J., & Durrett, G. (2019). Understanding dataset design choices for multi-hop reasoning. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4026–4032.

Chen, J., Lin, S.-t., & Durrett, G. (2019). Multi-hop question answering via reasoning chains. *arXiv preprint arXiv:1910.02610*.

Chen, L., Feng, Y., Huang, S., Qin, Y., & Zhao, D. (2014). Encoding relation requirements for relation extraction via joint inference. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 818–827. https://doi.org/10.3115/v1/P14-1077

Chen, X., Liang, C., Yu, A. W., Zhou, D., Song, D., & Le, Q. V. (2019). Neural symbolic reader: Scalable integration of distributed and symbolic representations for reading comprehension. *International Conference on Learning Representations*.

Chia, Y. K., Witteveen, S., & Andrews, M. (2019). Red dragon ai at textgraphs 2019 shared task: Language model assisted explanation generation. *arXiv preprint arXiv:1911.08976*.

Choi, E., Hewlett, D., Uszkoreit, J., Polosukhin, I., Lacoste, A., & Berant, J. (2017). Coarse-to-fine question answering for long documents. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 209–220.

Choi, Y., Breck, E., & Cardie, C. (2006). Joint extraction of entities and relations for opinion recognition. *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, 431–439. https://aclanthology.org/W06-1651

Choi, Y., & Cardie, C. (2009). Adapting a polarity lexicon using integer linear programming for domain-specific sentiment classification. *Proceedings of the 2009 conference on empirical methods in natural language processing*, 590–598.

Clark, P. (2015). Elementary school science and math tests as a driver for ai: Take the aristo challenge! *Twenty-Seventh IAAI Conference*.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., & Tafjord, O. (2018). Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Clark, P., Etzioni, O., Khot, T., Sabharwal, A., Tafjord, O., Turney, P. D., & Khashabi, D. (2016). Combining retrieval, statistics, and inference to answer elementary science questions. *AAAI*, 2580–2586.

Clark, P., Harrison, P., & Balasubramanian, N. (2013). A study of the knowledge base requirements for passing an elementary science test. *Proceedings of the 2013 workshop on Automated knowledge base construction*, 37–42.

Clark, P., Tafjord, O., & Richardson, K. (2021). Transformers as soft reasoners over language. *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, 3882–3890.

Dai, E., Sun, Y., & Wang, S. (2020). Ginger cannot cure cancer: Battling fake health news with a comprehensive data repository. *Proceedings of the International AAAI Conference on Web and Social Media*, *14*, 853–862.

Das, R., Godbole, A., Zaheer, M., Dhuliawala, S., & McCallum, A. (2019). Chains-of-reasoning at textgraphs 2019 shared task: Reasoning over chains of facts for explainable multi-hop inference. *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13)*, 101–117.

De Klerk, E. (2006). *Aspects of semidefinite programming: Interior point algorithms and selected applications* (Vol. 65). Springer Science & Business Media.

Demszky, D., Guu, K., & Liang, P. (2018). Transforming question answering datasets into natural language inference datasets. *arXiv preprint arXiv:1809.02922*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186.

Dhingra, B., Zaheer, M., Balachandran, V., Neubig, G., Salakhutdinov, R., & Cohen, W. W. (2020). Differentiable reasoning over a virtual knowledge base. *arXiv preprint arXiv:2002.10640*.

Ding, M., Zhou, C., Chen, Q., Yang, H., & Tang, J. (2019). Cognitive graph for multi-hop reading comprehension at scale. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2694–2703.

Djolonga, J., & Krause, A. (2017). Differentiable learning of submodular models. *Advances in Neural Information Processing Systems*, *30*, 1013–1023.

Donti, P. L., Amos, B., & Kolter, J. Z. (2017). Task-based end-to-end model learning in stochastic optimization. *arXiv preprint arXiv:1703.04529*.

D?Souza, J., Mulang, I. O., & Auer, S. (2019). Team svmrank: Leveraging feature-rich support vector machines for ranking explanations to elementary science questions. *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13)*, 90–100.

Du, X., & Cardie, C. (2018). Harvesting paragraph-level question-answer pairs from wikipedia. *arXiv preprint arXiv:1805.05942*.

Dua, D., Singh, S., & Gardner, M. (2020). Benefits of intermediate annotations in reading comprehension. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 5627–5634. https://www.aclweb.org/anthology/2020.acl-main.497

Dua, D., Wang, Y., Dasigi, P., Stanovsky, G., Singh, S., & Gardner, M. (2019). Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2368–2378.

Evans, J. S. B. (1984). Heuristic and analytic processes in reasoning. *British Journal of Psychology*, *75*(4), 451–468.

Evans, J. S. B. (2003). In two minds: Dual-process accounts of reasoning. *Trends in cognitive sciences*, *7*(10), 454–459.

Fang, Y., Sun, S., Gan, Z., Pillai, R., Wang, S., & Liu, J. (2020). Hierarchical graph network for multi-hop question answering. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 8823–8838.

Feldman, Y., & El-Yaniv, R. (2019). Multi-hop paragraph retrieval for open-domain question answering. *arXiv preprint arXiv:1906.06606*.

Feng, Y., Yu, M., Xiong, W., Guo, X., Huang, J., Chang, S., Campbell, M., Greenspan, M., & Zhu, X. (2020). Learning to recover reasoning chains for multi-hop question answering via cooperative games. *arXiv preprint arXiv:2004.02393*.

Ferreira, D., & Freitas, A. (2020a). Natural language premise selection: Finding supporting statements for mathematical text. *Proceedings of The 12th Language Resources and Evaluation Conference*, 2175–2182.

Ferreira, D., & Freitas, A. (2020b). Premise selection in natural language mathematical texts. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7365–7374.

Fodor, J. A. (1975). *The language of thought* (Vol. 5). Harvard university press.

Frazier, P. I. (2018). A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*.

Fried, D., Jansen, P., Hahn-Powell, G., Surdeanu, M., & Clark, P. (2015). Higher-order lexical semantic models for non-factoid answer reranking. *Transactions of the Association for Computational Linguistics*, *3*, 197–210.

Garg, S., Vu, T., & Moschitti, A. (2019). Tanda: Transfer and adapt pre-trained transformer models for answer sentence selection. *arXiv preprint arXiv:1911.04118*.

Geva, M., & Berant, J. (2018). Learning to search in long documents using document structure. *arXiv preprint arXiv:1806.03529*.

Gontier, N., Sinha, K., Reddy, S., & Pal, C. (2020). Measuring systematic generalization in neural proof generation with transformers. *Advances in Neural Information Processing Systems*, *33*, 22231–22242.

Gravina, A., Rossetto, F., Severini, S., & Attardi, G. (2018). Cross attention for selection-based question answering. *NL4AI@ AI* IA*, 53–62.

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2018). A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, *51*(5), 1–42.

Gupta, A., & Srinivasaraghavan, G. (2020). Explanation regeneration via multi-hop ILP inference over knowledge base. *Proceedings of the Graph-based Methods for Natural Language Processing (TextGraphs)*, 109–114. https://doi.org/10.18653/v1/2020.textgraphs-1.13

Gupta, N., Lin, K., Roth, D., Singh, S., & Gardner, M. (2020). Neural module networks for reasoning over text. *International Conference on Learning Representations*. https://openreview.net/forum?id=SygWvAVFPr

Gururangan, S., Swayamdipta, S., Levy, O., Schwartz, R., Bowman, S., & Smith, N. A. (2018). Annotation artifacts in natural language inference data. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 107–112.

Guu, K., Hashimoto, T. B., Oren, Y., & Liang, P. (2018). Generating sentences by editing prototypes. *Transactions of the Association for Computational Linguistics*, *6*, 437–450.

He, P., Liu, X., Gao, J., & Chen, W. (2020). Deberta: Decoding-enhanced bert with disentangled attention. *International Conference on Learning Representations*.

Helmberg, C. (2000). Semidefinite programming for combinatorial optimization.

Henderson, P., Hu, J., Romoff, J., Brunskill, E., Jurafsky, D., & Pineau, J. (2020). Towards the systematic reporting of the energy and carbon footprints of machine learning. *Journal of Machine Learning Research*, *21*(248), 1–43.

Holzinger, A., Biemann, C., Pattichis, C. S., & Kell, D. B. (2017). What do we need to build explainable ai systems for the medical domain? *arXiv preprint arXiv:1712.09923*.

Huang, L., Le Bras, R., Bhagavatula, C., & Choi, Y. (2019). Cosmos qa: Machine reading comprehension with contextual commonsense reasoning. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2391–2401.

Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2011). Sequential model-based optimization for general algorithm configuration. *International conference on learning and intelligent optimization*, 507–523.

Inoue, N., Stenetorp, P., & Inui, K. (2020). R4c: A benchmark for evaluating rc systems to get the right answer for the right reason. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 6740–6750.

Jansen, P. (2018). Multi-hop inference for sentence-level textgraphs: How challenging is meaningfully combining information for science question answering? *arXiv preprint arXiv:1805.11267*.

Jansen, P., Balasubramanian, N., Surdeanu, M., & Clark, P. (2016). What?s in an explanation? characterizing knowledge and inference requirements for elementary science exams. *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2956–2965.

Jansen, P., Sharp, R., Surdeanu, M., & Clark, P. (2017). Framing qa as building and ranking intersentence answer justifications. *Computational Linguistics*, *43*(2), 407–449.

Jansen, P., Thayaparan, M., Valentino, M., & Ustalov, D. (2021). Textgraphs 2021 shared task on multi-hop inference for explanation regeneration. *Proceedings of the Fifteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-15)*.

Jansen, P., & Ustalov, D. (2019). TextGraphs 2019 shared task on multi-hop inference for explanation regeneration. *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13)*, 63–77. https://doi.org/10.18653/v1/D19-5309

Jansen, P., Wainwright, E., Marmorstein, S., & Morrison, C. (2018). Worldtree: A corpus of explanation graphs for elementary science questions supporting multi-hop inference. *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Jhamtani, H., & Clark, P. (2020). Learning to explain: Datasets and models for identifying valid reasoning chains in multihop question-answering. *arXiv preprint arXiv:2010.03274*.

Jiang, Y., & Bansal, M. (2019a). Avoiding reasoning shortcuts: Adversarial evaluation, training, and model development for multi-hop qa. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2726–2736.

Jiang, Y., & Bansal, M. (2019b). Self-assembling modular networks for interpretable multi-hop reasoning. *arXiv preprint arXiv:1909.05803*.

Jiang, Y., Joshi, N., Chen, Y.-C., & Bansal, M. (2019). Explore, propose, and assemble: An interpretable model for multi-hop reading comprehension. *arXiv preprint arXiv:1906.05210*.

Johnson, J., Douze, M., & Jégou, H. (2019). Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*.

Karmarkar, N. (1984). A new polynomial-time algorithm for linear programming. *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, 302–311.

Karp, R. M. (1972). Reducibility among combinatorial problems. *Complexity of computer computations* (pp. 85–103). Springer.

Kaushik, D., & Lipton, Z. C. (2018). How much reading does reading comprehension require? a critical investigation of popular benchmarks. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 5010–5015.

Khashabi, D., Azer, E. S., Khot, T., Sabharwal, A., & Roth, D. (2019). On the capabilities and limitations of reasoning for natural language understanding. *arXiv preprint arXiv:1901.02522*.

Khashabi, D., Chaturvedi, S., Roth, M., Upadhyay, S., & Roth, D. (2018). Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 252–262.

Khashabi, D., Khot, T., Sabharwal, A., Clark, P., Etzioni, O., & Roth, D. (2016). Question answering via integer programming over semi-structured knowledge.

Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, 1145–1152.

Khashabi, D., Khot, T., Sabharwal, A., & Roth, D. (2018). Question answering as global reasoning over semantic abstractions. *Conference of Association for the Advancement of Artificial Intelligence*.

Khashabi, D., Min, S., Khot, T., Sabharwal, A., Tafjord, O., Clark, P., & Hajishirzi, H. (2020). Unifiedqa: Crossing format boundaries with a single qa system. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, 1896–1907.

Khot, T., Clark, P., Guerquin, M., Jansen, P., & Sabharwal, A. (2020). Qasc: A dataset for question answering via sentence composition. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(05), 8082–8090.

Khot, T., Sabharwal, A., & Clark, P. (2017). Answering complex questions using open information extraction. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. https://doi.org/10.18653/v1/p17-2049

Khot, T., Sabharwal, A., & Clark, P. (2019). What's missing: A knowledge gap guided approach for multi-hop question answering. *arXiv preprint arXiv:1909.09253*.

Klein, A., Falkner, S., Bartels, S., Hennig, P., & Hutter, F. (2017). Fast bayesian optimization of machine learning hyperparameters on large datasets. *Artificial Intelligence and Statistics*, 528–536.

Koomen, P., Punyakanok, V., Roth, D., & Yih, W.-t. (2005). Generalized inference with multiple semantic role labeling systems. *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, 181–184.

Kumar, S., & Talukdar, P. (2020). Nile: Natural language inference with faithful natural language explanations. *arXiv preprint arXiv:2005.12116*.

Kundu, S., Khot, T., Sabharwal, A., & Clark, P. (2019). Exploiting explicit paths for multi-hop reading comprehension. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2737–2747. https://doi.org/10.18653/v1/P19-1263

Kushner, H. J. (1964). A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*, *86*, 97–106.

Lai, G., Xie, Q., Liu, H., Yang, Y., & Hovy, E. (2017). RACE: Large-scale ReAding comprehension dataset from examinations. *Proceedings of the 2017 Conference*

on Empirical Methods in Natural Language Processing, 785–794. https://doi.org/10.18653/v1/D17-1082

Latcinnik, V., & Berant, J. (2020). Explaining question answering models through text generation. *arXiv preprint arXiv:2004.05569*.

Li, Y., & Clark, P. (2015). Answering elementary science questions by constructing coherent scenes using background knowledge. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2007–2012.

Liang, Y., Li, S., Yan, C., Li, M., & Jiang, C. (2021). Explaining the black-box model: A survey of local interpretation methods for deep neural networks. *Neurocomputing*, *419*, 168–182.

Lin, B. Y., Chen, X., Chen, J., & Ren, X. (2019). Kagnet: Knowledge-aware graph networks for commonsense reasoning. *arXiv preprint arXiv:1909.02151*.

Lin, Y., Ji, H., Liu, Z., & Sun, M. (2018). Denoising distantly supervised open-domain question answering. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1736–1745.

Liu, J., & Gardner, M. (2020). Multi-step inference for reasoning over paragraphs. *arXiv preprint arXiv:2004.02995*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Loboda, A. A., Artyomov, M. N., & Sergushichev, A. A. (2016). Solving generalized maximum-weight connected subgraph problem for network enrichment analysis. *International Workshop on Algorithms in Bioinformatics*, 210–221.

Lovász, L., & Schrijver, A. (1991). Cones of matrices and set-functions and 0-1 optimization. *SIAM JOURNAL ON OPTIMIZATION*, *1*, 166–190.

Lukasiewicz, T., Camburu, O., Shillingford, B., Blunsom, P., & Minervini, P. (2019). Make up your mind! adversarial generation of inconsistent natural language explanations. *arXiv preprint arXiv:1910.03065*.

Lv, S., Guo, D., Xu, J., Tang, D., Duan, N., Gong, M., Shou, L., Jiang, D., Cao, G., & Hu, S. (2019). Graph-based reasoning over heterogeneous external knowledge for commonsense question answering. *arXiv preprint arXiv:1909.05311*.

Mathew, B., Saha, P., Yimam, S. M., Biemann, C., Goyal, P., & Mukherjee, A. (2021). Hatexplain: A benchmark dataset for explainable hate speech detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(17), 14867–14875.

McCoy, T., Pavlick, E., & Linzen, T. (2019). Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3428–3448.

Mihaylov, T., Clark, P., Khot, T., & Sabharwal, A. (2018). Can a suit of armor conduct electricity? a new dataset for open book question answering. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2381–2391.

Mihaylov, T., & Frank, A. (2018). Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge. *arXiv preprint arXiv:1805.07858*.

Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, *267*, 1–38.

Min, S., Wallace, E., Singh, S., Gardner, M., Hajishirzi, H., & Zettlemoyer, L. (2019). Compositional questions do not necessitate multi-hop reasoning. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4249–4257.

Min, S., Zhong, V., Socher, R., & Xiong, C. (2018). Efficient and robust question answering from minimal context over documents. *arXiv preprint arXiv:1805.08092*.

Min, S., Zhong, V., Zettlemoyer, L., & Hajishirzi, H. (2019). Multi-hop reading comprehension through question decomposition and rescoring. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 6097–6109.

Mitra, A., Banerjee, P., Pal, K. K., Mishra, S., & Baral, C. (2019). How additional knowledge can improve natural language commonsense question answering? *arXiv preprint arXiv:1909.08855*.

Močkus, J. (1975). On bayesian methods for seeking the extremum. In G. I. Marchuk (Ed.), *Optimization techniques ifip technical conference novosibirsk, july 1–7, 1974* (pp. 400–404). Springer Berlin Heidelberg.

Ni, J., Zhu, C., Chen, W., & McAuley, J. (2019). Learning to attend on essential terms: An enhanced retriever-reader model for open-domain question answering. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 335–344.

Nie, Y., Wang, S., & Bansal, M. (2019). Revealing the importance of semantic retrieval for machine reading at scale. *arXiv preprint arXiv:1909.08041*.

Nishida, K., Nishida, K., Nagata, M., Otsuka, A., Saito, I., Asano, H., & Tomita, J. (2019). Answering while summarizing: Multi-task learning for multi-hop qa with evidence extraction. *arXiv preprint arXiv:1905.08511*.

Niu, Y., Jiao, F., Zhou, M., Yao, T., Xu, J., & Huang, M. (2020). A self-training method for machine reading comprehension with soft evidence extraction. *arXiv preprint arXiv:2005.05189*.

Norouzi, S., Fleet, D. J., & Norouzi, M. (2020). Exemplar vae: Linking generative models, nearest neighbor retrieval, and data augmentation. *Advances in Neural Information Processing Systems*, *33*.

Paige, C., & Saunders, M. (1982). An algorithm for sparse linear equations and sparse least squares: Acm transactions in mathematical software.

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). Bleu: A method for automatic evaluation of machine translation. *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 311–318.

Paulus, A., Rolınek, M., Musil, V., Amos, B., & Martius, G. (2021). Comboptnet: Fit the right np-hard problem by learning integer programming constraints. *International Conference on Machine Learning*, 8443–8453.

Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global vectors for word representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543. https://doi.org/10.3115/v1/D14-1162

Perez, E., Lewis, P., Yih, W.-t., Cho, K., & Kiela, D. (2020). Unsupervised question decomposition for question answering. *arXiv preprint arXiv:2002.09758*.

Pogančić, M. V., Paulus, A., Musil, V., Martius, G., & Rolinek, M. (2019). Differentiation of blackbox combinatorial solvers. *International Conference on Learning Representations*.

Punyakanok, V., Roth, D., Yih, W.-t., & Zimak, D. (2004). Semantic role labeling via integer linear programming inference. *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, 1346–1352. https://aclanthology.org/C04-1197

Qi, P., Lin, X., Mehr, L., Wang, Z., & Manning, C. D. (2019). Answering complex open-domain questions through iterative query generation. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2590–2602.

Qiu, B., Chen, X., Xu, J., & Sun, Y. (2019). A survey on neural machine reading comprehension. *arXiv preprint arXiv:1906.03824*.

Qiu, L., Xiao, Y., Qu, Y., Zhou, H., Li, L., Zhang, W., & Yu, Y. (2019). Dynamically fused graph network for multi-hop reasoning. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 6140–6150.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, *1*(8), 9.

Raiman, J., & Miller, J. (2017). Globally normalized reader. *arXiv preprint arXiv:1709.02828*.

Rajagopal, D., Tandon, N., Clarke, P., Dalvi, B., & Hovy, E. (2020). What-if i ask you to explain: Explaining the effects of perturbations in procedural text. *arXiv preprint arXiv:2005.01526*.

Rajani, N. F., McCann, B., Xiong, C., & Socher, R. (2019). Explain yourself! leveraging language models for commonsense reasoning. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4932–4942.

Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Reddy, S., Chen, D., & Manning, C. D. (2019). Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, *7*, 249–266.

Reimers, N., Gurevych, I., Reimers, N., Gurevych, I., Thakur, N., Reimers, N., Daxenberger, J., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*.

Robertson, S., Zaragoza, H. et al. (2009). The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, *3*(4), 333–389.

Ross, A. S., Hughes, M. C., & Doshi-Velez, F. (2017). Right for the right reasons: Training differentiable models by constraining their explanations. *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 2662–2670.

Roth, D., & Yih, W.-t. (2004). A linear programming formulation for global inference in natural language tasks. *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, 1–8. https://aclanthology.org/W04-2401

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, *1*(5), 206–215.

Saha, S., Ghosh, S., Srivastava, S., & Bansal, M. (2020). Prover: Proof generation for interpretable reasoning over rules. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 122–136.

Saha, S., Yadav, P., & Bansal, M. (2021). Multiprover: Generating multiple proofs for improved interpretability in rule reasoning. *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 3662–3677.

Sales, J. E., Freitas, A., & Handschuh, S. (2020). A user-centred analysis of explanations for a multi-component semantic parser. *International Conference on Applications of Natural Language to Information Systems*, 37–44.

Schlegel, V., Valentino, M., Freitas, A., Nenadic, G., & Batista-Navarro, R. (2020). A framework for evaluation of machine reading comprehension gold standards. *arXiv preprint arXiv:2003.04642*.

Schneider, S. (2011). *The language of thought: A new philosophical direction*. Mit Press.

Schrijver, A. (1998). *Theory of linear and integer programming*. John Wiley & Sons.

Schuff, H., Adel, H., & Vu, N. (2020). F1 is not enough! models and evaluation towards user-centered explainable question answering.

Seo, M., Kembhavi, A., Farhadi, A., & Hajishirzi, H. (2016). Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.

Shao, N., Cui, Y., Liu, T., Wang, S., & Hu, G. (2020). Is graph structure necessary for multi-hop reasoning? *arXiv preprint arXiv:2004.03096*.

Sharp, R., Surdeanu, M., Jansen, P., Valenzuela-Escárcega, M. A., Clark, P., & Hammond, M. (2017). Tell me why: Using question answering as distant supervision for answer justification. *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, 69–79.

Silva, V. S., Freitas, A., & Handschuh, S. (2019). Exploring knowledge graphs in an interpretable composite approach for text entailment. *Proceedings of the AAAI Conference on Artificial Intelligence*, *33*, 7023–7030.

Silva, V. S., Handschuh, S., & Freitas, A. (2018). Recognizing and justifying text entailment through distributional navigation on definition graphs. *Thirty-Second AAAI Conference on Artificial Intelligence*.

Sloman, S. A. (1996). The empirical case for two systems of reasoning. *Psychological bulletin*, *119*(1), 3.

Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, *25*, 2951–2959.

Song, K., Tan, X., Qin, T., Lu, J., & Liu, T.-Y. (2020). Mpnet: Masked and permuted pre-training for language understanding. *Advances in Neural Information Processing Systems*, *33*, 16857–16867.

Speer, R., Chin, J., & Havasi, C. (2017). Conceptnet 5.5: An open multilingual graph of general knowledge. *Thirty-First AAAI Conference on Artificial Intelligence*.

Srinivas, N., Krause, A., Kakade, S. M., & Seeger, M. W. (2009). Gaussian process bandits without regret: An experimental design approach. *CoRR*, *abs/0912.3995*. http://arxiv.org/abs/0912.3995

Stanovsky, G., Michael, J., Zettlemoyer, L., & Dagan, I. (2018). Supervised open information extraction. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 885–895. https://doi.org/10.18653/v1/N18-1081

Subramanian, S., Bogin, B., Gupta, N., Wolfson, T., Singh, S., Berant, J., & Gardner, M. (2020). Obtaining faithful interpretations from compositional neural networks. *arXiv preprint arXiv:2005.00724*.

Sun, K., Yu, D., Yu, D., & Cardie, C. (2019). Improving machine reading comprehension with general reading strategies. *Proceedings of the 2019 Conference of the North*. https://doi.org/10.18653/v1/n19-1270

Swanson, K., Yu, L., & Lei, T. (2020). Rationalizing text matching: Learning sparse alignments via optimal transport. *arXiv preprint arXiv:2005.13111*.

Tafjord, O., Dalvi, B., & Clark, P. (2021). Proofwriter: Generating implications, proofs, and abductive statements over natural language. *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 3621–3634.

Talmor, A., Herzig, J., Lourie, N., & Berant, J. (2019). Commonsenseqa: A question answering challenge targeting commonsense knowledge. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4149–4158.

Tandon, N., Dalvi, B., Sakaguchi, K., Clark, P., & Bosselut, A. (2019). Wiqa: A dataset for ?what if...? reasoning over procedural text. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 6078–6087.

Tenney, I., Xia, P., Chen, B., Wang, A., Poliak, A., McCoy, R. T., Kim, N., Van Durme, B., Bowman, S., Das, D., et al. (2019). What do you learn from context? probing for sentence structure in contextualized word representations. *7th International Conference on Learning Representations, ICLR 2019*.

Thapper, J., & Živný, S. (2017). The limits of sdp relaxations for general-valued csps. *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*.

Trivedi, H., Balasubramanian, N., Khot, T., & Sabharwal, A. (2020). Measuring and reducing non-multifact reasoning in multi-hop question answering. *arXiv preprint arXiv:2005.00789*.

Trivedi, H., Kwon, H., Khot, T., Sabharwal, A., & Balasubramanian, N. (2019). Repurposing entailment for multi-hop question answering tasks. *arXiv preprint arXiv:1904.09380*.

Tschiatschek, S., Sahin, A., & Krause, A. (2018). Differentiable submodular maximization. *arXiv preprint arXiv:1803.01785*.

Tu, M., Huang, K., Wang, G., Huang, J., He, X., & Zhou, B. (2019). Select, answer and explain: Interpretable multi-hop reading comprehension over multiple documents. *arXiv preprint arXiv:1911.00484*.

Valentino, M., Thayaparan, M., Ferreira, D., & Freitas, A. (2022). Hybrid autoregressive inference for scalable multi-hop explanation regeneration. *Proceedings of the AAAI Conference on Artificial Intelligence*, *36*(10), 11403–11411.

Valentino, M., Thayaparan, M., & Freitas, A. (2020). Explainable natural language reasoning via conceptual unification.

Valentino, M., Thayaparan, M., & Freitas, A. (2021). Unification-based reconstruction of multi-hop explanations for science questions. *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 200–211.

Vandenberghe, L., & Boyd, S. (1996). Semidefinite programming. *SIAM review*, *38*(1), 49–95.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 5998–6008.

Wang, C., Liang, S., Zhang, Y., Li, X., & Gao, T. (2019). Does it make sense? and why? a pilot study for sense making and explanation. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4020–4026.

Wang, H., Yu, D., Sun, K., Chen, J., Yu, D., McAllester, D., & Roth, D. (2019). Evidence sentence extraction for machine reading comprehension. *arXiv preprint arXiv:1902.08852*.

Wang, H., Yu, M., Guo, X., Das, R., Xiong, W., & Gao, T. (2019). Do multi-hop readers dream of reasoning chains? *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, 91–97.

Wang, P.-W., Donti, P., Wilder, B., & Kolter, Z. (2019). Satnet: Bridging deep learning and logical reasoning using a differentiable satisfiability solver. *International Conference on Machine Learning*, 6545–6554.

Wang, R., Tao, K., Song, D., Zhang, Z., Ma, X., Su, X., & Dai, X. (2020). R3: A reading comprehension benchmark requiring reasoning processes. *arXiv preprint arXiv:2004.01251*.

Weber, L., Minervini, P., Munchmeyer, J., Leser, U., & Rocktäschel, T. (2019). Nlprolog: Reasoning with weak unification for question answering in natural language. *arXiv preprint arXiv:1906.06187*.

Welbl, J., Stenetorp, P., & Riedel, S. (2018). Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, *6*, 287–302.

Williams, A., Nangia, N., & Bowman, S. R. (2017). A broad-coverage challenge corpus for sentence understanding through inference. https://doi.org/10.48550/ARXIV.1704.05426

Wilson, A., Fern, A., & Tadepalli, P. (2014). Using trajectory data to improve bayesian optimization for reinforcement learning. *The Journal of Machine Learning Research*, *15*(1), 253–282.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., & Brew, J. (2019). Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, *abs/1910.03771*. http://arxiv.org/abs/1910.03771

Wolfson, T., Geva, M., Gupta, A., Gardner, M., Goldberg, Y., Deutch, D., & Berant, J. (2020). Break it down: A question understanding benchmark. *Transactions of the Association for Computational Linguistics*, *8*, 183–198.

Wolsey, L. A. (2020). *Integer programming*. John Wiley & Sons.

Xie, Z., Thiem, S., Martin, J., Wainwright, E., Marmorstein, S., & Jansen, P. (2020). WorldTree v2: A corpus of science-domain structured explanations and inference patterns supporting multi-hop inference. *Proceedings of the 12th Language Resources and Evaluation Conference*, 5456–5473. https://www.aclweb.org/anthology/2020.lrec-1.671

Xu, W., Zhang, H., Cai, D., & Lam, W. (2021). Dynamic semantic graph construction and reasoning for explainable multi-hop science question answering. *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 1044–1056.

Yadav, V., Bethard, S., & Surdeanu, M. (2019a). Alignment over heterogeneous embeddings for question answering. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2681–2691.

Yadav, V., Bethard, S., & Surdeanu, M. (2019b). Quick and (not so) dirty: Unsupervised selection of justification sentences for multi-hop question answering. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2578–2589. https://doi.org/10.18653/v1/D19-1260

Yadav, V., Bethard, S., & Surdeanu, M. (2020). Unsupervised alignment-based iterative evidence retrieval for multi-hop question answering. *arXiv preprint arXiv:2005.01218*.

Yang, Y., Yih, W.-t., & Meek, C. (2015). Wikiqa: A challenge dataset for open-domain question answering. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2013–2018.

Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W. W., Salakhutdinov, R., & Manning, C. D. (2018). Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Ye, D., Lin, Y., Liu, Z., Liu, Z., & Sun, M. (2019). Multi-paragraph reasoning with knowledge-enhanced graph neural network. *arXiv preprint arXiv:1911.02170*.

Yoshida, Y. (2011). Optimal constant-time approximation algorithms and (unconditional) inapproximability results for every bounded-degree csp. *Proceedings of the forty-third annual ACM symposium on Theory of computing*, 665–674.

Yu, L., Hermann, K. M., Blunsom, P., & Pulman, S. (2014). Deep learning for answer sentence selection. *arXiv preprint arXiv:1412.1632*.

Zhang, X., Yang, A., Li, S., & Wang, Y. (2019). Machine reading comprehension: A literature review. *arXiv preprint arXiv:1907.01686*.

Zhang, Y., Dai, H., Toraman, K., & Song, L. (2018). Kgˆ 2: Learning to reason science exam questions with contextual knowledge graph embeddings. *arXiv preprint arXiv:1805.12393*.

Zhang, Z., Zhao, H., & Wang, R. (2020). Machine reading comprehension: The role of contextualized language models and beyond. *arXiv preprint arXiv:2005.06249*.

Zhao, C., Xiong, C., Rosset, C., Song, X., Bennett, P., & Tiwary, S. (2020). Transformer-xh: Multi-evidence reasoning with extra hop attention. *International Conference on Learning Representations*. https://openreview.net/forum?id=r1eIiCNYwS