UNIVERSITY
OF TECHNOLOGY
SYDNEY

Towards Robust and Interpretable Logical Reasoning in Machine Reading Comprehension

by Hao Huang

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

under the supervision of A/Prof. Guodong Long
A/Prof. Jing Jiang

University of Technology Sydney
Faculty of Engineering and Information Technology

Aug 2023

Certificate of Original Authorship

I, Hao Huang, declare that this thesis is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the Australian Artificial Intelligence Institute, Faculty of Engineering and Information Technology, at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Signature: Production Note:
Signature removed prior to publication.

Date: 3 April 2024

ABSTRACT

Towards Robust and Interpretable Logical Reasoning in Machine Reading Comprehension

by

Hao Huang

Natural Language Processing (NLP) has made significant strides using large pretrained language models in recent years. However, Natural Language Understanding (NLU) necessitates more profound understanding and reasoning capabilities that traditional NLP methods struggle to provide. This thesis concentrates on four aspects of augmentation of machine reading comprehension models: knowledge graph completion, procedural text understanding, temporal order extraction, and autodebiasing. Collectively, these components contribute to robust and interpretable logical reasoning in machine reading comprehension step by step.

A knowledge graph is a structured representation of knowledge, typically in the form of a directed graph, where entities are represented as nodes and their relationships are represented as edges. Knowledge graph completion is a task in natural language processing and machine learning that involves predicting missing facts or relationships between entities in a knowledge graph. Translating embedding approaches offer advantages such as lightweight structure, high efficiency, and excellent interpretability. In particular, when extended to complex vector space, they can handle various relation patterns, including symmetry, antisymmetry, inversion, and composition. Nevertheless, previous translating embedding approaches defined in complex vector space suffer from two main issues: 1) limited representing and modeling capacities due to the translation function's rigorous multiplication of two complex numbers; and 2) unaddressed embedding ambiguity caused by one-to-many relations. This thesis introduces our published work that features a relation-adaptive translation function built upon a novel weighted product in a complex space. Our model's weights are learnable, relation-specific, and independent of embedding size.

Procedural text understanding aims to track entities' states (e.g., creation, movement, destruction) and locations as mentioned in a given paragraph. Effectively tracking these requires capturing the rich semantic relations between entities, actions, and locations in the paragraph. While recent works have made considerable progress, they focus on leveraging inherent constraints or incorporating external knowledge for state prediction, largely overlooking the given paragraph's rich semantic relations. We introduce our published novel approach (Real) for procedural text understanding, where we build a general framework to systematically model entity-entity, entity-action, and entity-location relations using a graph neural network. We further develop algorithms for graph construction, representation learning, and state and location tracking.

Temporal reading comprehension (TRC) involves reading a free-text passage and answering temporal ordering questions. Precise question understanding is crucial for temporal reading comprehension. To address this, we propose a novel reading comprehension approach with precise question understanding. Specifically, we embed a temporal ordering question into two vectors to capture the referred event and the temporal relation. This fine-grained representation offers two benefits: first, it enables a better understanding of the question by focusing on different elements of a question; second, it provides good interpretability when evaluating temporal relations. Furthermore, we incorporate an auxiliary contrastive loss for representation learning of temporal relations, aiming to distinguish relations with subtle but critical differences.

Despite the success of large pre-trained language models in natural language understanding benchmarks, recent studies indicate that these models often rely on superficial features or shortcuts to make predictions. In this thesis, we explore an automatic method for progressively detecting and filtering biased data to train a robust debiased model for natural language understanding tasks. Diverging from previous debiasing methods that concentrate on human-predefined biases or biases captured by limited-capacity bias-only models, we introduce a novel debiasing framework called Bias-Progressive Auto-Debiasing. This framework is based on two observations: i) a higher proportion of biased samples in training data results in a more biased model, and ii) a more biased model exhibits greater confidence in predicting biases. Our framework progressively trains a bias-only model using the most biased samples identified in the previous epoch, thereby ensuring a more biased model and ultimately leading to a robust debiased model.

Key Words. Natural Language Understanding, Knowledge Graph Completion, Procedural Text, Graph Reasoning, Temporal Relation, Debiasing

Directed by Prof. Guodong Long,
School of Computer Science, Faculty of Engineering and Information Technology

Acknowledgements

I would like to begin by expressing my profound gratitude to Professor Guodong Long, my esteemed supervisor. His invaluable guidance, enduring patience, and unwavering support were instrumental throughout the course of this thesis. Additionally, my sincere appreciation goes to my co-supervisor, Dr. JingJiang, and Professor Chengqi Zhang. Their keen insights, constructive feedback, and steadfast encouragement have been pivotal at every stage of this research.

I wish to extend special thanks to my senior colleague, Tao Shen. His collaboration, friendship, and invaluable wisdom have been cornerstones of my academic journey. Equally, my gratitude is extended to Xiubo Geng, my research intern mentor, who offered a unique perspective, continuous support, and invaluable mentorship that enriched my research experience. Furthermore, I'm thankful to my diligent teammates: Yang Li, Zhuowei Wang, Jie Ma, Wensi Tang, Zonghan Wu, and Xueping Peng, who have all played significant roles in my academic pursuits. I'd also like to acknowledge the guidance of Fangfang Li, my mentor during my internship, for her invaluable insights.

On a more personal note, my heartfelt thanks go to my parents, Minxin Han and Qinghua Huang. Their unwavering faith in me, even in the face of challenges, has been my guiding light. Lastly, to my wife, Manqing Dong, mere words cannot capture the depth of my gratitude. Your unyielding understanding, patience, and love have truly been the foundation of this significant milestone.

Hao Huang
Sydney, Australia, 2023

List of Publications

Published Conference Paper:

C-1 **Hao Huang**, Guodong Long, Tao Shen, Jing Jiang and Chengqi Zhang: RatE: Relation-Adaptive Translating Embedding for Knowledge Graph Completion. Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020: 556-567 (CORE A)

C-2 **Hao Huang**, Xiubo Geng, Pei Jian, Guodong Long and Daxin Jiang: Reasoning over Entity-Action-Location Graph for Procedural Text Understanding. Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021: 5100-5109 (CORE A*)

C-3 **Hao Huang**, Xiubo Geng, Guodong Long and Daxin Jiang: Understand before Answer: Improve Temporal Reading Comprehension via Precise Question Understanding. Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022: 375-384 (CORE A)

C-4 **Hao Huang**, Tianyi Zhou, Tao Shen, Jing Jiang, Guodong Long and Chengqi Zhang: Auto-Debiasing by Boosting a Biased Model. Under Review

Contents

List of Figures

Abbreviation

Natural Language Processing - NLP

Natural Language Understanding - NLU

Artificial Intelligence - AI

Machine reading comprehension - MRC

Long Short Term Memory - LSTM

Gate Recurrent Unit - GRU

Bidirectional Encoder Representations from Transformer - BERT

Large Language Models - LLMs

Knowledge Graph - KG

Pre-trained Language Models - PLMs

Graph Neural Networks - GNNs

Temporal Reading Comprehension - TRC

Chapter 1

Introduction

1.1 Background

Machine Reading Comprehension (MRC) is a fundamental pillar of natural language processing (NLP). Its core aim is to enable machines to read, comprehend, and respond to human language as intuitively as we do. In the early stages of NLP, our methods were basic, primarily centered on matching words without truly delving into their underlying context. However, the field witnessed rapid and momentous evolution. From those rudimentary beginnings, we transitioned to sophisticated models like Recurrent Neural Networks, including variants like Long short-term memory (LSTM) and Gate Recurrent Unit (GRU). These models excel at remembering long-term dependencies and discerning the intricate semantic relationships embedded within text sequences. A pivotal shift occurred with the advent of the Transformer architecture. Leveraging the self-attention mechanism, these transformer-based models efficiently recognize relationships across a text, bi-directionally, and are impressively scalable. This capability is epitomized in large language models like Bidirectional Encoder Representations from Transformers (BERT) and Generative Pre-trained Transformer (GPT), which have redefined the benchmarks for MRC. In the present, colossal models such as GPT, backed by extensive training on vast text datasets, showcase profound language understanding. Tracing this journey from simple word matching to deep textual comprehension highlights the monumental advancements we've achieved in narrowing the chasm between machine capabilities and human linguistic intuition. In the realm of Machine Reading Comprehension

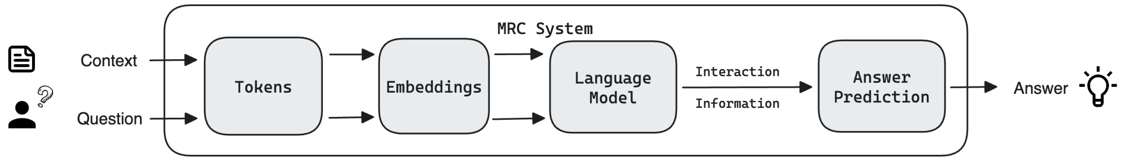


Figure 1.1 : An overview of MRC.

(MRC), Large Language Models (LLMs) have emerged as game-changers. Their extensive training over vast textual corpora provides them a nuanced understanding of context, enabling them to interpret complex passages and answer queries with high accuracy. Their deep architectures, rooted in the Transformer model, excel at discerning intricate relationships within a text, making them adept at distinguishing between closely related facts and drawing inferences. The scalable self-attention mechanism inherent in LLMs allows them to weigh relevant portions of a text dynamically, ensuring that even subtle cues aren't missed. Furthermore, LLMs' capacity for transfer learning means that they can be fine-tuned on specific MRC tasks or datasets, delivering performance that often rivals or exceeds specialized models.

However, despite the remarkable advancements, there remain innate challenges. One of the most persistent is ensuring models can reliably perform logical reasoning on textual data. Such reasoning isn't just about parsing sentences; it involves synthesizing information, drawing inferences, and applying knowledge contextually. This becomes even more intricate due to the inherent complexity and ambiguity that natural languages possess. Furthermore, the absence of structured, consistent data to anchor this reasoning process exacerbates the issue. Adding to these concerns, LLMs, with their vast and intricate architectures, often present themselves as "black boxes," making it challenging to interpret why they produce specific outputs. This lack of interpretability can be problematic in applications where understanding the model's decision-making process is crucial. Moreover, ethical concerns arise due to the potential biases embedded within LLMs. Being trained on vast swaths of internet text, these models might inadvertently perpetuate and amplify societal biases they encounter, adding another layer of complexity to their reliable and unbiased deployment. To tackle these challenges, this thesis zeroes in on four pivotal facets of MRC: knowledge graph completion, procedural text understanding, temporal order comprehension, and auto-debiasing. Integrating these methods enhances the robustness, reliability, and fairness of AI systems in processing and understanding natural language. The methods of knowledge graph completion and auto-debiasing serve as general enhancements for LLMs, while procedural text understanding and temporal order comprehension offer targeted modifications for specific types of MRC tasks.

Knowledge graph completion is a subfield of artificial intelligence that involves inferring missing facts in a knowledge graph (KG). Knowledge graphs are a structured representation of real-world entities and their relationships, which can be used to support logical reasoning in MRC. However, incomplete or inaccurate knowledge graphs can significantly affect the performance of MRC models. A knowledge graph refers to a collection of interlinked entities, which is usually formatted as a set of triples. A triple is represented as a head entity linked to a tail entity by a relation, which is written as (*head, relation, tail*) or (*h, r, t*). Large-scale knowledge graphs, such as Freebase (Bollacker et al., 2008) and WordNet (Miller, 1995), containing structured information, have been leveraged to support a broad spectrum of natural language processing (NLP) tasks, e.g., question answering (Hao et al., 2017), recommender system (Zhang et al., 2016), relation extraction (Min et al., 2013), etc. Nonetheless, the human-curated, real-world knowledge graphs often suffer from incompleteness or sparseness problem (Toutanova et al., 2015), which inevitably hurts the performance of downstream tasks. Hence, how to auto-complete knowledge graphs becomes a popular problem in both research and industry communities. LLMs, with their capability to understand and generate human-like text, can be paired with KGs for advanced question answering. If there's a missing piece of information in the KG, the LLM can be used to predict or infer it, making the Q&A system more robust and accurate.

Procedural text understanding is another important aspect of MRC that involves understanding the steps involved in a given procedure or process. This requires not only identifying the relevant entities and their relationships, but also understanding the temporal and causal relationships between them. Procedural text understanding is a challenging task due to the diversity and complexity of procedural texts. Procedural text often consists of a sequence of sentences describing processes, such as a phenomenon in nature (e.g., how sedimentary rock forms) (Dalvi et al., 2018) or instructions to complete a task (e.g., the recipe of Mac and Cheese) (Bosselut et al., 2018). Given a paragraph and its participant entities, the task of procedural text understanding is to track the states (e.g., create, move, destroy) and locations (a span in the text) of the entities. Compared with traditional machine reading task, which mainly focuses on the static relations among entities, procedural text understanding is more challenging since it involves discovering complex temporal-spatial relations among various entities from the process dynamics. Therefore, we verify that graphs inherently represent relationships. By translating text into a graph, LLMs can better perform relational reasoning, understanding how different entities relate and interact with one another.

Temporal order comprehension is a subtask of MRC that involves identifying the chronological order of events described in a text. This is important for tasks such as event prediction and story understanding, but can be challenging due to the complex and ambiguous nature of natural language. Temporal order extraction requires the ability to recognize temporal cues, such as temporal adverbs and verb tenses, and to infer the temporal relationships between events. Understanding temporal relationships between events in a passage is essential for natural language understanding (Wang et al., 2019b; Dong et al., 2019). Temporal reading comprehension (TRC) (Ning et al., 2020) is a natural way to study temporal relations since natural language questions are flexible to capture divergent temporal relations (Zhou et al., 2021). By forcing the model to differentiate between similar temporal sequences, contrastive learning can enhance the LLM's ability to recognize subtle differences in sequences. This helps the model better capture the nuances and patterns inherent in temporal data, which can also generalize to wilder tasks.

Auto-debiasing is a technique that aims to mitigate the effects of biases in MRC models. Biases can arise from various sources, such as the training data or the model architecture, and can significantly affect the accuracy and fairness of MRC models. Auto-debiasing involves identifying and correcting biases in the model, by either modifying the training data or introducing additional constraints in the model architecture. In the last decade, deep representation learning has demonstrated its general capabilities across a broad spectrum of tasks and made significant progress on natural language understanding datasets such as GLUE (Wang et al., 2018) and SuperGLUE (Wang et al., 2019a). However, recent studies (Poliak et al., 2018; Gururangan et al., 2018; McCoy et al., 2019) reveal that the models tend to capture *dataset biases* (i.e., the superficial clues such as word overlaps and negative words) to make predictions, rather than learning from the underlying features. Such an issue becomes the main barrier to the models' reliability in deployment, particularly when it comes to out-of-distribution generalization. Moreover, the issue remains for the recent large-scale pre-trained models with generic representations. As such, reducing the impact of *dataset biases* becomes the key challenge in learning robust natural language understanding (NLU) models. For users and organizations, an

LLM that has undergone debiasing becomes more trustworthy as it ensures that outputs are not reflecting harmful stereotypes or biases. By removing biases, models can potentially generalize better to various tasks as they aren't overfitting to biased nuances in the training data.

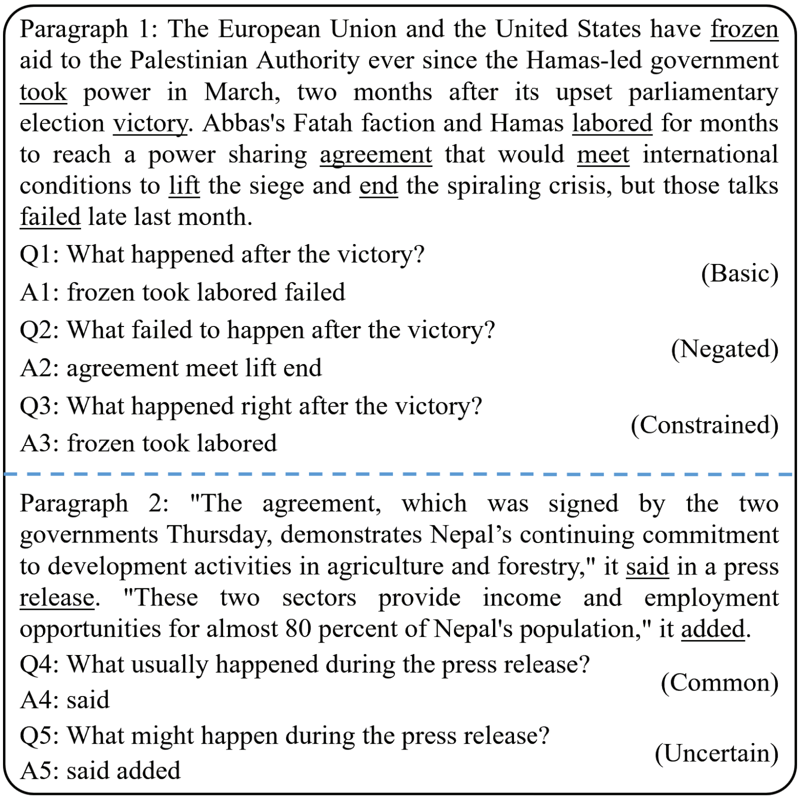
1.2 Research Problems

Current Machine Reading Comprehension (MRC) models frequently encounter difficulties when dealing with intricate logical reasoning, sequential interpretations, temporal comprehension, and inherent biases present in training data. These challenges lead to a deficient understanding of textual data and potential misinterpretations. Recognizing these limitations, our aim is to develop an MRC system that can robustly and interpretably navigate these complexities, leveraging the aspects previously delineated. Each existing model presents unique limitations concerning each of these aspects. We will delve into the corresponding research problems in the ensuing paragraphs.

Knowledge graph completion Current trans-based graph embedding approaches with complex embeddings are vulnerable to the following two issues. On the one hand, although approaches solely in complex vector space are equipped with high interpretability for various relation patterns, they are limited by the expressive power of standard product/add of two complex numbers. To improve, QuatE (Zhang et al., 2019a) introduces quaternion hypercomplex vector space with semantic matching, at the cost of both interpretability and computational overheads, but the improvement is still marginal. On the other hand, embedding ambiguity problem, which means different entities are assigned with similar embeddings, cannot be explicitly handled by existing trans-based approaches (e.g., TransE and RotatE). It is mainly caused by the propagation of applying a translation function to one-to-many relations for optimizing $\forall \boldsymbol{t} = g(\boldsymbol{h}, \boldsymbol{r})$.

Procedural text understanding To effectively track the states and locations of entities, it is crucial to systematically model rich relations among various concepts in the paragraph, including entities, actions, and locations. Three types of relations are of particular interest. First, mentions of the same entity in different sentences are related. The inherent relation among these mentions may provide clues for a model to generate consistent predictions about the entity. For example, the entity *electrical pulses* are mentioned in two sentences "*The retina's rods and cones convert it to electrical pulses. The optic nerve carries electrical pulses through the optic canal.*". Connecting its two mentions in two sentences helps to infer its location in the first sentence using the second sentence's information. Second, detecting connections between an entity and the corresponding actions helps to make state predictions more accurate. Take the sentence "*As the encased bones decay, minerals seep in replacing the organic material.*" as an example. The entity *bone* is related to *decay* which indicates the state *destroy*, while it is not connected to *seep* indicating the state *move*. Given the relation between *bone* and *decay*, it is easier for the model to predict the state of *bone* as *destroy*, instead of being misled by the action *seep*. Last, when the state or location of one entity changes, it may impact all associated entities. For example, in sentence "*trashbags are thrown into trashcans.*", *trashbags* are associated with *trashcans*. Then, in the following sentence "*The trashcan is emptied by a large trash truck.*", although *trashbags* are not explicitly mentioned, their locations are changed by the association with *trashcan*.

Temporal order comprehension Figure 1.2 shows several examples of temporal reading comprehension, where given a free-text passage, a system is required to answer temporal questions like "*What usually happened during the press release?*".



Paragraph 1: The European Union and the United States have frozen aid to the Palestinian Authority ever since the Hamas-led government took power in March, two months after its upset parliamentary election victory. Abbas's Fatah faction and Hamas labored for months to reach a power sharing agreement that would meet international conditions to lift the siege and end the spiralling crisis, but those talks failed late last month.

Q1: What happened after the victory?
A1: frozen took labored failed
(Basic)

Q2: What failed to happen after the victory?
A2: agreement meet lift end
(Negated)

Q3: What happened right after the victory?
A3: frozen took labored
(Constrained)

Paragraph 2: "The agreement, which was signed by the two governments Thursday, demonstrates Nepal's continuing commitment to development activities in agriculture and forestry," it said in a press release. "These two sectors provide income and employment opportunities for almost 80 percent of Nepal's population," it added.

Q4: What usually happened during the press release?
A4: said
(Common)

Q5: What might happen during the press release?
A5: said added
(Uncertain)

Figure 1.2 : Examples of temporal reading comprehension. Temporal relations are diverse: Q1-Q5 list examples of possible varieties of temporal relations. Small changes in the question might lead to substantially divergent semantics: replacing *usually* in Q4 with *might* in Q5 leads to different answers. Related events are underlined in the passage.

A natural solution for temporal ordering understanding is to compare each candidate answer and the referred event in the question and classify their temporal relation into several pre-defined categories, e.g., (UzZaman et al., 2013) defines 13 possible relations such as *after, ends, equal to.* Nonetheless, since temporal relationships vary greatly, it is almost impossible to enumerate all possible relationships. Figure 1.2 shows several divergent varieties of temporal relations: one might query about *plain after* in Q1, *negated after* in Q2, *constrained after* in Q3, etc. Similarly, a question might query about *usually happen* in Q4, *might happen*, or other relations.

Moreover, creating sufficient labels for all such relations is costly and poses great challenges for real-world applications. Therefore, the classification-based approach is incompetent to handle the flexible relations in temporal reading comprehension.

Auto-debiasing Early debiasing methods often relied heavily on manual analysis by human experts (He et al., 2019; Clark et al., 2019; Mahabadi et al., 2020) to identify potential biases in specific datasets and define the most likely bias types. However, these experience-dependent methods can be time-consuming and may not cover all types of biases. To address these challenges, recent studies have focused on developing automatic and dataset-agnostic debiasing methods for NLU that can cover a wider range of bias types. These methods typically involve training a *bias-only model* to implicitly or explicitly detect biased samples, which are then down-weighted in the training of a *debiased model.* Therefore, the critical problem reduces to train a *bias-only model.* In previous works, two heuristic assumptions are commonly used to train the bias-only model. The first is the "*weak-model*" assumption, which posits that models with lower capacity (e.g., Bag-of-words models or Tiny-BERT) are more likely to learn from the shallow heuristics of datasets and thus result in a bias-only model (Sanh et al., 2020). The second is the "*small-data*" assumption, which states that a model is prone to fitting shortcuts or biased features in the dataset during its early training stages (Utama et al., 2020b). However, the assumptions used to train a bias-only model in previous works are uncertain and have many uncontrollable factors. It is difficult to define how weak the model should be or how small the dataset should be, resulting in redundant hyperparameters. Additionally, the bias-only model is inevitably fed with normal or robust samples due to both i) the unknown dataset-specific biasing sample proportion and ii) the randomness of model selection or data sampling. These uncontrollable factors can lead to a less-biased bias-only model, negatively impacting the learning of the debiased model. Thereby, our goal is to develop a stable, automatic method for training a better biased model that is agnostic to the dataset, bias type, model size, and data scale.

1.3 Thesis Organization

This thesis is organized as follows:

- Chapter 2: This chapter explores the existing literature, spotlighting landmark studies within various domains such as Machine Reading Comprehension (MRC) tasks, Knowledge Graph Completion, Procedural Text Understanding, Temporal Order Comprehension, and Auto-Debiasing.
- Chapter 3: This chapter introduces our novel technique for Knowledge Graph Completion, addressing limited representation and ambiguity in translating embedding. Our embedding method integrated knowledge graph structure and semantic context. Detailed experimentation supporting this approach is presented at the end of the chapter.
- Chapter 4: This chapter unveils our comprehensive framework designed to methodically model relationships between entities, actions, and locations using a Graph Neural Network. A thorough evaluation of this approach, bolstered by detailed experimentation, will be presented at the conclusion of the chapter.
- Chapter 5: This chapter presents a novel approach to reading comprehension with precise question comprehension. It incorporates an auxiliary contrastive loss mechanism, with the objective of discerning relations. A thorough explanation of the experiments is provided at the end of this chapter.
- Chapter 6: This chapter explores an automatic method for progressively detecting and filtering biased data to train a robust debiased model for natural

language understanding tasks. The details of the experiments are given in the last part of the chapter.

- Chapter 7: This chapter offers a concise recapitulation of the entire research undertaking, while also outlining potential avenues for future exploration and study.

Chapter 2

Literature Review

The subsequent literature review encompasses a range of topics that are intimately linked with the development of a robust and interpretable Machine Reading Comprehension (MRC) system.

Knowledge Graph Completion Unlike semantic matching graph embedding approaches (Nickel et al., 2011; Dettmers et al., 2017; Balazevic et al., 2019; Zhang et al., 2019a) require additional overheads to score a triple, this work is in line with trans-based graph embedding approaches that employ an efficient translation function defined in a latent space. TransE (Bordes et al., 2013) is the most representative trans-based approach, which embeds entities/relations in real vector space and utilizes the relations as translations. It optimizes score function towards "$\boldsymbol{h} + \boldsymbol{r} = \boldsymbol{t}$". Several recent trans-based approaches (Wang et al., 2014; Lin et al., 2015; Ji et al., 2015; Ebisu and Ichise, 2018) can be viewed as extensions of TransE. More recently, RotatE (Sun et al., 2019), as a state-of-the-art trans-based approach, represents the entities/relations in complex vector space and formulates the translating process as a rotation in complex space.

Negative Sampling Also related to this work, many negative sampling methods (Cai and Wang, 2018; Sun et al., 2019) are proposed to effectively learn structured knowledge. KBGAN (Cai and Wang, 2018) uses knowledge graph embedding model as a negative sample generator to fool the main embedding model (i.e., the discriminator in GANs). In contrast, self-adversarial learning (Sun et al., 2019) scores a certain number of uniformly-sampled negative samples based on current model, and utilizes the scores to perform a weighted loss function. Lastly, this work is also related to using prior knowledge in graphs for training. Type-constraint method (Krompaß et al., 2015), which is based on local closed-world assumptions, corrupts heads (or tails) from relation-specific domain (or range).

Machine Reading Comprehension Machine reading comprehension (MRC) (Rajpurkar et al., 2016, 2018; Shen et al., 2018a,b; Li et al., 2020) has attracted much attention in recent years. Traditional solutions to MRC tasks focus on utilizing the interaction information between questions and passages via attention-based structures (Kadlec et al., 2016; Dhingra et al., 2017). Later on, pre-trained language models (PLMs), e.g., BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and XLNet (Yang et al., 2019), have been widely used for MRC tasks. With the sheer scale of parameters and the pretraining strategies, PLMs capture more knowledge from the context and have shown outstanding performance on traditional MRC benchmarks. For more challenging MRC tasks which introduce multi-hop reasoning (Yang et al., 2018), numerical reasoning (Dua et al., 2019), etc., the generic PLMs become not applicable. Recent efforts use graph-based reasoning approaches (Chen et al., 2020) or define specific pretraining training techniques (Raffel et al., 2020) to solve the above challenges. However, existing MRC approaches still struggle for the temporal reading comprehension task due to the lack of temporal relation understanding (Ning et al., 2020). Hence, we propose a novel question answering approach with precise question understanding to tackle this challenge.

Procedural Text Understanding Compared with early-stage models (Henaff et al., 2017; Seo et al., 2017), recent progress in the procedural text understanding task is mainly made on ensuring the prediction's consistency or injecting external knowledge. Various approaches (Dalvi et al., 2018; Gupta and Durrett, 2019; Amini et al., 2020) have been proposed to predict consistent state sequence. For example, NCET (Gupta and Durrett, 2019) tracks the entity in a continuous space and leverages a conditional random field (CRF) to keep a consistent prediction sequence. Other models inject knowledge from external data sources to complement missing knowledge. ProStruct (Tandon et al., 2018) introduces commonsense constraints to refine the probability space, while KOALA (Zhang et al., 2020) leverages Bert Encoder pre-trained on related corpus from Wiki, and injects the ConceptNet (Speer et al., 2017) knowledge. Besides, a few models (Das et al., 2019; Dalvi et al., 2019) are proposed to build graphs on the procedural text. For instance, KG-MRC (Das et al., 2019) constructs dynamic knowledge graphs between entities and locations. However, these methods can not systematically capture the relations among entities, actions, and locations, and entity-action and entity-entity relations are ignored.

Graph Reasoning in Language Understanding Graph-based reasoning methods (Zeng et al., 2020; Zhong et al., 2020; Zheng and Kordjamshidi, 2020) are widely used in natural language understanding tasks to enhance performance. For example, Zeng et al. (2020) constructs a double graph design for the document-level Relation Extraction (RE) task, Zhong et al. (2020) constructs the retrieved evidence sentences as a graph for Fact-Checking task. Compared with these works, the entity-action-location graph in our approach copes better with procedural text understanding task since it precisely defines concepts we are concerned within the task and captures the rich and expressive relations among them.

Temporal Ordering Reasoning Traditional temporal order reasoning tasks (UzZaman et al., 2013; Cassidy et al., 2014; Ning et al., 2018), are often formulated as relation extraction tasks. Given the context passage, the target is to classify the relation between every two events from a predefined relation set, e.g., UzZaman et al. (2013) defines 13 possible relations such as *after, ends, equal to.* Existing solutions can be roughly classified into two categories. The first category focuses on developing the structure of the encoder to capture more temporal information. For example, Cheng et al. (2020) add up a GRU-based dynamically updating structure upon the outputs of the common BERT sentence encoder. The second category focuses on joint learning with external knowledge or some specific constraints. For instance, Ning et al. (2019) significantly improve the extraction performance by joint training temporal and causal relations.

Free-text Temporal Ordering However, the success of the existing approaches is limited to the formulation of the traditional temporal order reasoning tasks, where the events and the candidate temporal relation set are fixed. However, the fixed candidate relation set cannot cover all temporal relations in our daily uses. The most recent released dataset, TORQUE (Ning et al., 2020), formulates temporal ordering reasoning as a machine reading comprehension task. Given a context passage, we need to answer a free-text question about the temporal relations in the context passage. The task is much analogous to our real-world tasks and is more challenging – we need to automatically identify the events and the relations in the free-text question to retrieve the answers from the context passage. To the best of our knowledge, we are the very first to address this challenge.

Bias in Datasets *Dataset bias* is inevitable in most human-crafted datasets (Wang et al., 2018, 2019a), such bias could be simple word co-occurrence (Gururangan et al., 2018), negation words (Utama et al., 2020b), or overlap relation between premise and hypothesis in natural language inference tasks (McCoy et al., 2019). Recent studies reveal that models can outperform random guesses by merely utilizing these biases as shortcuts (Tsuchiya, 2018; Poliak et al., 2018; Nie et al., 2020; Saxon et al., 2022). However, the performance of fine-tuned models drops significantly when tested on filtered bias-free datasets or new complex samples. Therefore, debiasing methods are essential for obtaining robust models that can capture underlying semantics.

Data-centric Debiasing Methods Existing debiasing methods can be broadly classified as *data-centric* and *model-centric* methods. *Data-centric* methods focus on improving the quality of the training data by either removing biased samples (Le Bras et al., 2020) or generating new unbiased samples (Zhang et al., 2019b; Wu et al., 2022). For example, Le Bras et al. (2020) use adversarial filtering to remove dataset biases and train the model on filtered datasets, while Zhang et al. (2019b) generate additional training samples through controlled word exchange and back-translation, with human checks for fluency and paraphrase judgment. However, researchers have shown that newly constructed datasets may not be entirely bias-free and may introduce significant overhead, highlighting the need for robust debiasing algorithms.

Model-centric Debiasing Methods Model-centric methods share a common idea of building a bias-only model to identify biased instances and then reducing their importance during training using methods such as i) example reweighting (Schuster et al., 2019), i.e., down weighting the biased samples, ii) confidence regularization (Utama et al., 2020a), i.e., force the model to be less confident on the biased samples, and iii) product-of-experts (He et al., 2019; Mahabadi et al., 2020), which introduces the output of the bias-only model to the training objective function. For building bias-only models, current methods are based on observations by Sanh et al. (Sanh et al., 2020) and Utama et al. (Utama et al., 2020b). Sanh et al. (2020) find that a model with limited capacity (e.g., TinyBERT) can be more biased than larger models and train a TinyBERT on the whole dataset as the bias-only model. Utama et al. (2020b) find that a model is more biased when trained on a smaller dataset at an early stage and train a BERT-base with a small fraction of the training dataset as the bias-only model. However, both methods cannot guarantee a strongly biased model as the bias-only models are not trained on bias-only datasets – the former utilizes the entire dataset, and the latter randomly selects the subset, which will still bring general knowledge to the bias-only model. In this work, we propose a new bias-progressive auto-debiasing framework that ensures a stronger bias-only model and a robust debiased model.

Chapter 3

Relation-Adaptive Translating Embedding for Knowledge Graph Completion

3.1 Introduction

Human-curated, real-world knowledge graphs often suffer from incompleteness or sparseness problem (Toutanova et al., 2015), which inevitably hurts the performance of downstream tasks. Hence, how to auto-complete knowledge graphs becomes a popular problem in both research and industry communities.

For this purpose, many light-weight graph embedding approaches (Bordes et al., 2013; Yang et al., 2015; Sun et al., 2019) have been proposed. Unlike costly graph neural networks (GNNs) (Schlichtkrull et al., 2018), these approaches use low-dimensional embeddings to represent the entities and relations, and capture their relationships via semantic matching or geometric distance. Specifically, the approaches with semantic matching, e.g., DistMult (Yang et al., 2015) and QuatE (Zhang et al., 2019a), use a matching function $f(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t})$ that operates on whole triple to directly derive its plausibility score. In contrast, the approaches with geometric distance, e.g., TransE (Bordes et al., 2013) and RotatE (Sun et al., 2019), first apply a translation function to head entity and relation for a new embedding in latent space and then measure a distance from the new embedding to tail entity, i.e., $f(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}) = -||g(\boldsymbol{h}, \boldsymbol{r}) - \boldsymbol{t}||_p$. Empirically, the latter, namely *trans-based approach*, usually has higher efficiency and superior performance on link prediction than the former. Based on translating process, it also offers better interpretability of the graph embeddings and relation modeling (Sun et al., 2019).

Recently, some trans-based graph embedding approaches, e.g., RotatE (Sun et al., 2019), go beyond real vector space. They represent the entities and relations in complex vector space, and define the translation function on complex vectors. Empowered by the properties of arithmetic operations (e.g., product) in complex space, the translation function can easily capture relation patterns of symmetry (e.g., *marriage*), antisymmetry (e.g., *father*), inversion(e.g., *hypernym* vs. *hyponym*) and composition (e.g., $mother \wedge husband \rightarrow father$). Compared to those defined in real vector space, these approaches improve model's capability in handling a variety of relation patterns and achieve state-of-the-art performance.

Nevertheless, current trans-based graph embedding approaches with complex embeddings are vulnerable to the following two issues. On the one hand, although approaches solely in complex vector space are equipped with high interpretability for various relation patterns, they are limited by the expressive power of standard product/add of two complex numbers. To improve, QuatE (Zhang et al., 2019a) introduces quaternion hypercomplex vector space with semantic matching, at the cost of both interpretability and computational overheads, but the improvement is still marginal. On the other hand, embedding ambiguity problem, which means different entities are assigned with similar embeddings, cannot be explicitly handled by existing trans-based approaches (e.g., TransE and RotatE). It is mainly caused by the propagation of applying a translation function to one-to-many relations for optimizing $\forall \boldsymbol{t} = g(\boldsymbol{h}, \boldsymbol{r})$.

To alleviate both issues above, we propose a novel **R**elation-**a**daptive **t**ranslating **E**mbedding (RatE) (Huang et al., 2020) approach for knowledge graph completion. As an extension of the trans-based embedding approach RotatE, our proposed RatE inherits the capability to handle various relation patterns, and further presents a light-weight yet effective relation-adaptive translation function. Specifically, the function is composed of a novel element-wise *weighted product* defined in complex

Type	**Model**	**Score Function**	**Space**	**Sym.**	**Antisym.**	**Inv.**	**Comp.**	**Disambiguation**
Semantic matching	DistMult	$\langle \boldsymbol{r}, \boldsymbol{h}, \boldsymbol{t} \rangle$	$\mathbb{R}^d$	✓	✗	✗	✗	-
	ComplEx	$\mathrm{Re}(\langle \boldsymbol{r}, \boldsymbol{h}, \bar{\boldsymbol{t}} \rangle)$	$\mathbb{C}^d$	✓	✓	✓	✗	-
	QuatE	$\boldsymbol{h} \otimes \boldsymbol{r}^{\triangleleft} \cdot \boldsymbol{t}$	$\mathbb{H}^d$	✓	✓	✓	✗	-
Trans-based	TransE	$-\|\|\boldsymbol{h} + \boldsymbol{r} - \boldsymbol{t}\|\|$	$\mathbb{R}^d$	✗	✓	✓	✓	✗
	RotatE	$-\|\|\boldsymbol{h} \circ \boldsymbol{r} - \boldsymbol{t}\|\|_1$	$\mathbb{C}^d$	✓	✓	✓	✓	✗
	RatE	$-\|\|\boldsymbol{h} \odot_{\boldsymbol{W}^{(r)}} \boldsymbol{r} - \boldsymbol{t}\|\|_1$	$\mathbb{C}^d$	✓	✓	✓	✓	✓

Table 3.1 : A brief comparison of semantic matching and trans-based graph embedding approaches, where a check mark denotes the model is equipped with the corresponding property. "Sym.", "Antisym.", "Inv." and "Comp." are abbreviations of relation patterns of symmetry, antisymmetry, inversion and composition respectively. For a trans-based graph embedding model, "Disambiguation" denotes whether the model explicitly handles embedding ambiguity problem as detailed in §3.2.5. And, $\langle \cdot \rangle$ denotes generalized dot product, $\circ$ denotes element-wise (Hadamard) complex product, $\otimes$ denotes element-wise Hamilton product, $^{\triangleleft}$ denotes normalization of a vector, and $\odot_{\boldsymbol{W}}$ denotes our proposed weighted product defined in Eq.(3.2).

vector space, where the weights are learnable, relation-specific and independent to embedding dimension. Rather than rigorous complex number product in RotatE and QuatE, RatE provides a more flexible way – either the resulting real or imaginary part is a weighted sum of the product on every pair of numbers respectively from the two complex number arguments (i.e., real or imaginary part). Hence, RatE only requires *eight* more scalar parameters each relation than baseline RotatE, which is much less than the embedding dimension by one or two orders of magnitude. Through relation-adaptive translation function, the proposed approach empirically promotes the capacity of modeling translation process and embedding ambiguity problem, while preserves most interpretability to handle various relation patterns.

We also propose a novel local-cognitive negative sampling method, by integrating type-constraint training technique (Krompaß et al., 2015) with self-adversarial learning (Sun et al., 2019). The former leverages prior knowledge in graph during training and samples negative head (tail) entities from relation-specific domain (range), which is limited by the hard sampling criterion and suffers from graph sparseness. By comparison, the latter scores a certain number of uniformly-sampled negative samples based on current model, and uses the normalized scores as weights for the loss function. It hence depends heavily on an incompletely-trained model. Thus, we integrate them for their mutual benefits: besides using a self-adversarial loss, our method leverages prior knowledge to weaken the effect of current model.

3.2 Approach

This section begins with a definition of link prediction task for knowledge graph completion, followed by an introduction to a baseline RotatE. Then, we propose a novel relation-adaptive translation function to compose the final relation-adaptive translating embedding approach. Then, we present an efficient negative sampling method by integrating the merits of two previous sampling strategies. Lastly, we demonstrate the capability of our proposed model in alleviating embedding ambiguity problem.

3.2.1 Link Prediction

Formally, a knowledge graph $\mathcal{G} = \{\mathcal{E}, \mathcal{R}\}$ consists of a set of triples $(h,\ r,\ t)$, where $h, t \in \mathcal{E}$ are head and tail entities respectively while $r \in \mathcal{R}$ is the relation between them. Given a head h (or tail t) entity and a relation r, the goal of link prediction is to find the most accurate tail t (or head h) from $\mathcal{E}$ to make the new triple $(h,\ r,\ t)$ plausible in the knowledge graph $\mathcal{G}$. In a graph embedding approach, each entity/relation is assigned with an embedding vector, and a triple is denoted as $(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t})$. To tackle link prediction, a scoring function $f(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t})$ is presented to derive the plausibility score for each triple candidate. Especially in a trans-based approach, the score function is formulated as $f(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}) = -||g(\boldsymbol{h}, \boldsymbol{r}) - \boldsymbol{t}||_p$ where $g(\cdot)$ denotes a translation function.

3.2.2 Baseline: RotatE

RotatE is a state-of-the-art trans-based graph embedding approach in complex vector space. Motivated by Euler's identity, its translating process is formulated as a relation-specific rotation of the head's embedding vector. RotatE in complex space can be viewed as a natural extension of vanilla TransE in real vector space, aiming to support the relation pattern of symmetry. Specifically, RotatE represents both entities $\mathcal{E}$ and relations $\mathcal{R}$ in complex vector space $\mathbb{C}^d$, and defines relation's embedding as a rotation by constraining the modulus of each dimension to 1. And its translation function $g(\boldsymbol{h}, \boldsymbol{r})$ is simply fulfilled by a Hadamard product (i.e., element-wise, denoted as "$\circ$") in complex vector space, i.e., $g(\boldsymbol{h}, \boldsymbol{r}) = \boldsymbol{h} \circ \boldsymbol{r}$. Therefore, the scoring function in RotatE is written as

$$f(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}) = -||\boldsymbol{h} \circ \boldsymbol{r} - \boldsymbol{t}||_1, \text{ where } \boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t} \in \mathbb{C}^d \text{ and } \forall |r_i| = 1. \tag{3.1}$$

Note, the p-norm of a complex vector $\boldsymbol{v}$ is defined as $||\boldsymbol{v}||_p = \sqrt[p]{\sum |v_i|^p}$.

3.2.3 Relation-Adaptive Translating Embedding

Based on the baseline, we propose a trans-based graph embedding approach, named as **R**elation-**a**daptive **t**ranslating **E**mbedding (RatE). It extends complex number product to a novel *weighted product* in complex space, where the weights are learnable and relation-specific. The weighted product is defined as

$$o = u \otimes_{\boldsymbol{W}} v = (a + bi) \otimes_{\boldsymbol{W}} (c + di) = \boldsymbol{W}_{1,:} \boldsymbol{s}^{(u,v)} + \boldsymbol{W}_{2,:} \boldsymbol{s}^{(u,v)} i, \tag{3.2}$$
$$\text{where, } o, u, v \in \mathbb{C}, \boldsymbol{W} \in \mathbb{R}^{2\times 4} \text{ and } \boldsymbol{s}^{(u,v)} = [ac; ad; bc; bd] \in \mathbb{R}^4.$$

Here, $\boldsymbol{W}$ denotes a learnable weight matrix and will be updated during training for a specific target. Standard complex number product is its special case when $\boldsymbol{W} = [[1,0,0,-1];[0,1,1,0]]$. Hence, empowered by the learnable weights, the weighted product promotes the ability to implicitly capture arithmetic or geometrical relationships in complex space when adapted into a data-driven neural model.

Then, the proposed weighted product is readily integrated with RotatE to compose a novel relation-adaptive translation function. That is

$$\tilde{\boldsymbol{t}} := g(\boldsymbol{h}, \boldsymbol{r}) = \boldsymbol{h} \odot_{\boldsymbol{W}^{(r)}} \boldsymbol{r}, \text{ where, } \forall i: \ \tilde{t}_i = h_i \otimes_{\boldsymbol{W}^{(r)}} r_i, \ |r_i| = 1. \tag{3.3}$$

$\boldsymbol{h}, \boldsymbol{r} \in \mathbb{C}^d$ are the embeddings of head entity and relation respectively, and $\odot_{\boldsymbol{W}^{(r)}}$ denotes *element-wise weighted product* where the weights are specified for each relation $r \in \mathcal{R}$. Based on this translation function, we formulate the score function of relation-adaptive translating embedding as

$$s^{(h,r,t)} := f(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}) = -||\boldsymbol{h} \odot_{\boldsymbol{W}^{(r)}} \boldsymbol{r} - \boldsymbol{t}||_1, \tag{3.4}$$

where $s^{(h,r,t)} \in \mathbb{R}$ is the resulting score of the triple (h,r,t) to measure its plausibility. As both the graph embeddings and the translation function are defined in complex vector space and learnable during training, our proposed RatE is a generic formulation of previous trans-based approaches. In other words, the approaches like RotatE and TransE are special cases of RatE, so our approach makes the best of deep neural network and promotes the representing capacity of translating paradigm. This is achieved by increasing only *eight* learnable parameters for each relation, which are fewer than the relation's embedding size by one or two orders of magnitude. Moreover, besides handling the four relation patterns (i.e., symmetry, antisymmetry, inversion and composition), the proposed RatE also reduces the effect of embedding ambiguity (detailed at the end of this section). It is also noteworthy that although the integration above is based on RotatE, the proposed weighted product is compatible with any complex or hypercomplex embedding approach (e.g., QuatE).

3.2.4 Negative Sampling and Optimization

The way to conduct negative sampling can significantly affect the performance of a graph embedding approach (Cai and Wang, 2018; Sun et al., 2019) because contrasting a challenging negative sample against the corresponding positive one is more effective for learning structured knowledge. Formally, given an arbitrary correct triple $x = (h, r, t) \in \mathcal{G}^{(tr)}$, negative sampling aims at corrupting its either head or tail entity to get a wrong triple $x' = (h', r, t)$ or (h, r, t'), where $x' \notin \mathcal{G}^{(tr)}$. $\mathcal{G}^{(tr)}$ denotes the knowledge graph to train an embedding model. Note, we only exhibit tail corruption for a clear elaboration in the following, and head corruption is also considered in our implementation.

We first introduce two popular sampling strategies in the following. Type-constraint training technique (Krompaß et al., 2015) presents a new link prediction setting based on local closed-world assumptions – the entities to corrupt a triple only come from a relation-specific entity set during both training and test. We only take this idea in training phase to introduce prior knowledge and provide strong distractors. Particularly, for a triple (h, r, t), the candidate set of tail corruptions is

$$\mathcal{E}^{(h,r,t)} = \{t' \in \mathcal{E} | \exists e \in \mathcal{E} : (e, r, t') \in \mathcal{G}^{(tr)} \wedge (h, r, t') \notin \mathcal{G}^{(tr)}\}. \tag{3.5}$$

However, sampling only in this set, $\mathcal{E}^{(h,r,t)}$, suffers from not only graph sparseness by local closed-world assumptions but also information loss of other corrupting entities. The other entities are denoted as

$$\bar{\mathcal{E}}^{(h,r,t)} = \{t' \in \mathcal{E} | t' \notin \mathcal{E}^{(h,r,t)} \wedge (h, r, t') \notin \mathcal{G}^{(tr)}\}. \tag{3.6}$$

In contrast, self-adversarial negative sampling (Sun et al., 2019) applies triple scoring function to a certain number of uniformly-sampled wrong triples, and each $f(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}')$ represents its difficulty to current embedding model. It then uses the normalized scores as the weights in loss function to perform a self-adversarial training. However, this sampling strategy depends heavily on current embedding model.

Then, we propose a novel local-cognitive negative sampling method by integrating them to complement each other. Our integration is non-trivial, where a dynamic coefficient[*] $\gamma \in [0, 1]$ is used to control the proportion of negative samples from $\mathcal{E}^{(h,r,t)}$ or $\bar{\mathcal{E}}^{(h,r,t)}$. In particular, a certain number n of wrong triples is first sampled for each triple $x = (h, r, t) \in \mathcal{G}^{(tr)}$. To achieve this, we conduct a uniform sampling individually in $\mathcal{E}^{(h,r,t)}$ and $\bar{\mathcal{E}}^{(h,r,t)}$, which respectively produce $\mathcal{N}$ containing γn samples and $\bar{\mathcal{N}}$ containing $(1 - \gamma)n$ samples. Then we optimize the proposed embedding model by minimizing

$$\mathcal{L} = \mu ||\boldsymbol{W}^{(r)}||_1 - \log \sigma(\lambda + f(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t})) - \sum\nolimits_{(h,r,t') \in \mathcal{N} \cup \bar{\mathcal{N}}} \beta^{(h,r,t')} \log \sigma(-f(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}') - \lambda), \tag{3.7}$$

$$\text{where } \beta^{(h,r,t')} = \exp f(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}') / \sum\nolimits_{(h,r,t'') \in \mathcal{N} \cup \bar{\mathcal{N}}} \exp f(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}''). \tag{3.8}$$

μ is weight decay of L1 regularization and set to 0.01 without tuning. Lastly, we update the coefficient γ at the end of every training epoch by

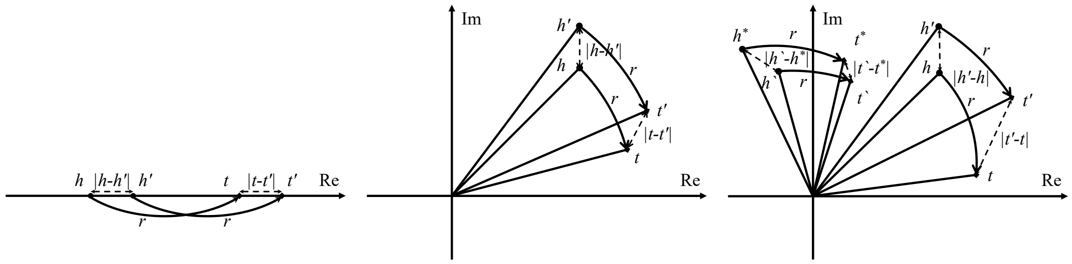
$$\gamma \leftarrow \frac{1}{|\mathcal{G}^{(tr)}|} \sum\nolimits_{\mathcal{G}^{(tr)}} 1 / \left(1 + \frac{\sum_{(h,r,t') \in \bar{\mathcal{N}}} \exp f(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}') / |\bar{\mathcal{N}}|}{\sum_{(h,r,t') \in \mathcal{N}} \exp f(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}') / |\mathcal{N}|} \right). \tag{3.9}$$

Here γ inclines to the candidate set with more challenging negative samples, which is determined by all wrong triples sampled in the previous epoch. In summary, our sampling method employs a self-adversarial training loss, and leverages prior knowledge to weaken the effect of current model.

3.2.5 Embedding Disambiguation

Embedding ambiguity here refers to similar embeddings assigned to distinct entities. In a trans-based graph embedding approach, it is usually caused by one-to-many (i.e., a kind of non-injective) relations in knowledge graphs. Specifically, given

[*] We initialize γ with 0.5 and empirically find the initialization value barely affects final performance.



(a) TransE (b) RotatE (c) RatE

Figure 3.1 : Toy examples – applying translation functions of TransE, RotatE and RatE to (h_i, r_i) for the resulting t_i. Note that 1) dimension index i is omitted, and 2) TransE is defined in real space whereas RotatE/RatE is defined in complex space.

a set of triples $\{(h, r, t^1), \ldots, (h, r, t^M)\}$ as an example of one-to-many relations, invoking a translation function directly defined in real or complex space makes the model optimize toward $\forall \boldsymbol{t}^j = g(\boldsymbol{h}, \boldsymbol{r})$ and inevitably results in similar tail embeddings. Because one-to-many relations are ubiquitous in a knowledge graph, e.g., *has_part* in WordNet, the embedding ambiguity problem will deteriorate and propagate through the graph. Fortunately, the proposed RatE is able to alleviate this problem by cutting off the propagation.

To intuitively demonstrate RatE's capability in embedding disambiguation by stopping the propagation, we respectively illustrate toy examples of TransE, RotatE, and our proposed RatE in Figure 3.1. It is observed that given two head entities with similar embeddings, their similarity will be preserved in corresponding tail entities after applying the same relation, not to mention the relation r possibly being a one-to-many relation. The triple scoring function built upon geometric distance may hardly discriminate such subtle differences in the space and thus negatively affects the quality of predictions. In principle, compared to rigid transformation in RotatE and TransE, the proposed RatE with weighted product shares a similar inspiration with projective transformation and changes the distance between the tail entities according to spatial positions of the head entities. Consequently, besides increasing the distance between the tail entities to disambiguate entity embeddings, RatE could also decrease the distance for better support of many-to-one relations. A rigorous proof of these properties is provided in Appendix A.

3.3 Experiment

3.3.1 Experimental Setting

Dataset	# Entity	# Relation	# Training	# Validation	# Test
WN18	40,943	18	141,442	5,000	5,000
FB15k	14,951	1,345	483,142	50,000	59,071
WN18RR	40,943	11	86,835	3,034	3,134
FB15k-237	14,541	237	272,115	17,535	20,466

Table 3.2 : Statistics of four benchmark datasets of link prediction.

Dataset. We employ four widely-used link prediction benchmark, WN18, FB15K, WN18RR and FB15K-237, whose statistics are summarized in Table 3.2. Note, Toutanova et al., (Toutanova et al., 2015) find that both WN18 and FB15K suffer from direct link problem caused by most test triples (e^1, r^1, e^2) can be found in the training or valid set with another relation, e.g., (e^1, r^2, e^2) or (e^2, r^2, e^1).

- WN18 (Bordes et al., 2013) is extracted from WordNet (Miller, 1995), a knowledge graph composed of English phrases and lexical relations between them.
- FB15k (Bordes et al., 2013) is extracted from Freebase (Bollacker et al., 2008), which is a large-scale knowledge graph consisting real-world named entities and their relationships.

- FB15k-237 (Toutanova et al., 2015) is a subset of FB15k by 1) removing near-duplicate and inverse triples, and 2) filtering out the direct links to avoid data leakage.
- WN18RR (Dettmers et al., 2017) is a subset of WN18 following the same processes as FB15k-237.

Training Setting. The ranges of the hyper-parameters for grid search are elaborated in the following. Embedding dimension $d \in \{250, 500, 1000\}$, batch size $\in \{512, 1024, 2048\}$, and fixed margin $\lambda \in \{6, 9, 12, 18\}$. By following previous works, all entities and relation embeddings are randomly initialized under uniform distribution. The initialization range of entities is $[-\lambda/d, +\lambda/d]$ for both real and imaginary parts, and the initialization range of relations is $[0, 2\pi]$ with $|\boldsymbol{r}| = \mathbf{1}$ in complex space. Our model is implemented using PyTorch on a single Titan V GPU. We use minibatch SGD with Adam optimizer, where the learning rate is set to 5×10^{-5} without decay.

Evaluation Metrics. Following Bordes et al. (Bordes et al., 2013), we use "*filtered*" setting to calculate evaluation metrics during test: In either head or tail entity corruption, all correct triples in train/dev/test except the current oracle test triple are removed to avoid affecting rank. Given all candidate triples ranked according to the score function $f(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t})$, we use the standard evaluation metrics on link prediction tasks: 1) *mean rank* (MR) to describe the mean rank of the oracle test triples, 2) *mean reciprocal rank* (MRR), and 3) *Hits@N* (N=1, 3, 10) to denotes the ratio of the oracle test triples ranked in top-N.

Comparative Approach. We compare RatE with several strong graph embedding approaches, especially the trans-based approaches to which RatE belongs. In particular, for trans-based approaches, we mainly consider TransE (Bordes et al., 2013) in real space and RotatE (Sun et al., 2019) in complex space. For semantic matching approaches, we consider DistMult (Yang et al., 2015), HolE (Nickel et al., 2016), ComplEx (Trouillon et al., 2016), ConvE (Dettmers et al., 2017) and QuatE (Zhang et al., 2019a). For most approaches, we copy results from the original paper or (Sun et al., 2019) except explanations.

3.3.2 Evaluation on Link Prediction

Method	**WN18**					**FB15k**				
	MR	MRR	Hits@10	Hits@3	Hits@1	MR	MRR	Hits@10	Hits@3	Hits@1
TransE	-	.495	.943	.888	.113	-	.463	.749	.578	.297
DistMult	655	.797	.946	-	-	42	.798	.893	-	-
HolE	-	.938	.949	.945	.930	-	.524	.739	.613	.402
ComplEx	-	.941	.947	.945	.936	-	.692	.840	.759	.599
ConvE	374	.943	.956	.946	.935	51	.657	.831	.723	.558
RotatE	309	.949	.959	.952	.944	40	.797	.884	.830	**.746**
QuatE	388	.949	.960	**.954**	.941	41	.770	.878	.821	.700
RatE	**180**	**.950**	**.962**	.953	**.944**	**24**	**.810**	**.898**	**.859**	.724

Table 3.3 : Link prediction results on WN18 and FB15k. The results of QuatE are reported without type-constraint.

Link prediction results on the four datasets are shown in Table 3.3 and Table 3.4. It is observed that the proposed RatE is able to achieve new state-of-the-art results in terms of most metrics compared to previous graph embedding approaches. Overall, compared with the baseline model RotatE, RatE merely employs several additional parameters to deliver significant improvement. To the best of our knowledge, RotatE is previous the best trans-based graph embedding approach and belongs to the same category as RatE. RatE also outperforms previous state-of-the-art semantic

Method	WN18RR					FB15k-237				
	MR	MRR	Hits@10	Hits@3	Hits@1	MR	MRR	Hits@10	Hits@3	Hits@1
TransE	3384	.226	.501	-	-	357	.294	.465	-	-
DistMult	5110	.430	.490	.440	.390	254	.241	.419	.263	.155
ComplEx	5261	.440	.510	.460	.410	339	.247	.428	.275	.158
ConvE	4187	.430	.520	.440	.400	244	.325	.501	.356	.237
RotatE	3340	.476	.571	.492	.428	177	.338	.533	.375	.241
QuatE	3472	.481	.564	.500	.436	176	.311	.495	.342	.221
RatE	**2860**	**.488**	**.590**	**.506**	**.441**	**172**	**.344**	**.541**	**.382**	**.261**

Table 3.4 : Link prediction results on WN18RR and FB15k-237. Values in bold denote the best results.

matching graph embedding approach, QuatE, which is defined in hypercomplex space and requires more computational overheads.

Specifically, since WN18 and FB15k suffer from the direct link problem as detailed above, it is observed that the baselines and our proposed RatE obtain comparable results in all metrics. For example, Dettmers et al. (Dettmers et al., 2017) find that using a rule-based model to learn the inverse relations achieves competitive results on WN18RR. This explains why our improvement is marginal in these two datasets. Moreover, since WN18RR and FB15k-237 are presented to solve the problem in WN18 and FB15k respectively, the evaluation results on WN18RR and FB15k-237 are more canonical to measure the capability in link prediction. As shown in Table 3.4, the proposed RatE brings a more noticeable improvement in contrast to previous approaches.

3.3.3 Ablation Study

We conduct an extensive ablation study in Table 3.5 to verify the effectiveness of each proposed part. We first replace the relation-adaptive translation function with

Method	**MR**	**MRR**	**Hits@10**	**Hits@3**	**Hits@1**
RatE full	2860	.488	.590	.506	.441
RatE w/o relation-adaptive	3278	.478	.579	.498	.432
RatE w/o weighted product	3115	.479	.576	.492	.432
RatE w/o $\boldsymbol{W}^{(r)}$ L1 reg	2921	.482	.584	.499	.435
RatE w/o negative sampling	3180	.471	.564	.478	.428
RatE w/o ALL	3450	.465	.556	.476	.410

Table 3.5 : Ablation study on WN18RR.

a shared weighted product among all relations (i.e., "RatE w/o relation-adaptive"), and observe a performance drop. And the weighted product further degenerates to standard complex product (i.e., RatE w/o weighted product), which only results in a slight drop. This suggests the proposed weighted product should be coupled with relation-adaptation to maximize its effectiveness. Then, removing L1 regularization of $\boldsymbol{W}^{(r)}$ in Eq.(3.7) and the proposed local-cognitive negative sampling leads to 0.6% and 2.6% Hits@10 drops respectively. Note "RatE w/o negative sampling" denotes using a uniform negative sampling method instead of our proposed local-cognitive negative sampling. Lastly, when removing all the proposed parts, the model is equivalent to its baseline RotatE without self-adversarial negative sampling, which results in inferior performance.

3.3.4 Analysis of Relation-Adaptive Translation Function

A major difference between RatE and previous trans-based graph embedding approaches (e.g., RotatE) is that a learnable relation-adaptive translation function is used in RatE to capture the translating relationship. To measure the expressive power of RatE, it is significant to investigate the learned weights in each relation-specific weighted product. As shown in Table 3.6, the L1 norm of learned $\boldsymbol{W}^{(r)}$ for

Relation Pattern	**Relation Name**	$\|\|\boldsymbol{W}^{(r)}\|\|_1$	**RatE**	**RotatE**	**TransE**
Symmetry	*verb_group*	2.3	**0.98**	0.97	0.87
	derivationally_related_form	2.5	**0.97**	**0.97**	0.93
	also_see	2.3	0.70	**0.73**	0.59
Antisymmetry	*instance_hypernym*	6.1	**0.56**	0.54	0.22
	synset_domain_topic_of	3.3	**0.49**	**0.49**	0.19
	member_of_domain_usage	6.6	**0.50**	0.49	0.42
	member_of_domain_region	6.1	**0.48**	0.45	0.35
	member_meronym	8.7	**0.54**	0.38	0.04
	has_part	8.1	**0.40**	0.35	0.04
	hypernym	7.1	**0.30**	0.27	0.02
Micro Mean	-	-	**0.59**	0.57	0.38

Table 3.6 : Test performance in Hits@10 regarding different relation patterns and the corresponding relations on WN18RR. $||\boldsymbol{W}^{(r)}||_1$ is used to measure the complexity of the proposed relation-adaptive translation function. Since only three triples with relation "*similar_to*" appear in the test set of WN18RR, we omit this relation.

symmetric relation is obviously less than that of antisymmetric relation. In particular, with the redundancy of complex number product removed, RatE preserves the ability to handle symmetric relations and achieves competitive results. For example, $\boldsymbol{W}^{(r)} = [[1.0, 0.1, 0.0, 0.1]; [0.0, 0.1, 1.0, 0.0]]$ is learned for relation "*verb_group*". In contrast, RatE tends to construct expressively powerful translation function for antisymmetric relations and achieves much better performance across these relations than previous models.

3.3.5 Performance on Non-Injective Relations

By following Sun et al. (Sun et al., 2019), we also evaluate the proposed RatE on different types including one injective relation type (i.e., one-to-one) and three non-injective relation types (i.e., one-to-many, many-to-one and many-to-many).

	Tail Prediction (Hits@10)				**Head Prediction (Hits@10)**			
Relation Type	1-to-1	1-to-M	M-to-1	M-to-M	1-to-1	1-to-M	M-to-1	M-to-M
TransE (Bordes et al., 2013)	.879	.671	.964	.910	.894	**.972**	.567	.880
RotatE (Sun et al., 2019)	.923	.713	.961	.922	.922	.967	.602	.893
RatE*	**.926**	**.801**	**.968**	**.924**	**.927**	.971	**.724**	**.895**

Table 3.7 : Performance on FB15k regarding different relation types, including injective (i.e., 1-to-1) and non-injective (e.g., 1-to-M) relations. *We replace the proposed local-cognitive negative sampling in RatE with self-adversarial one from RotatE.

As shown in Table 3.7, although RatE delivers similar Hits@10 values to RotatE on the injective relation type, it significantly surpasses both TransE and RotatE on the non-injective relation types. The improvements are especially substantial in 1-to-M relation (+8.8%) on tail prediction and M-to-1 (+12.2%) on head prediction, which verifies RatE's capability in handling one-to-many relations. Coupled with the theoretical proof in §3.2.5, this also indirectly verifies that RatE is able to alleviate the embedding ambiguity problem posted by one-to-many relations.

Negative Sampling Method	**WN18RR**					**FB15k-237**				
	MR	MRR	Hits@10	@3	@1	MR	MRR	Hits@10	@3	@1
Uniform	3180	.471	.564	.478	.428	224	.320	.525	.374	.220
Self-adversarial (Sun et al., 2019)	3114	.480	.576	.481	.433	177	.339	.536	.375	.244
Local-cognitive w/o self-adv loss	3094	.479	.577	.489	.434	180	.340	.538	.374	.241
Local-cognitive (ours)	**2860**	**.488**	**.590**	**.506**	**.441**	**172**	**.344**	**.541**	**.382**	**.261**

Table 3.8 : Performance of RatE with different negative sampling methods.

3.3.6 Analysis of Negative Sampling

As negative sampling is crucial for a model to learn structured knowledge, we evaluate RatE with different negative sampling methods. "Local-cognitive w/o self-

Method	**WN18RR**					**FB15k-237**					$\#\boldsymbol{\theta}$
	MR	MRR	Hits@10	@3	@1	MR	MRR	Hits@10	@3	@1	
TuckER (Balazevic et al., 2019)	-	.470	.526	.482	**.443**	-	**.358**	**.544**	**.394**	**.266**	$d_e^2 d_r$
RatE (ours)	**2860**	**.488**	**.590**	**.506**	.441	**172**	.344	.541	.382	.261	$8\|\mathcal{R}\|$

Table 3.9 : Performance comparison between TuckER and RatE on WN18RR/FB15k-237. "$\#\boldsymbol{\theta}$" denotes the number of learnable parameters only for scoring, where d_e and d_r are the embedding sizes of entity and relation respectively.

adv loss" can be viewed as only using prior knowledge from local closed-world assumptions (Krompaß et al., 2015). The experimental results shown in Table 3.8 demonstrate that compared with uniform sampling, both self-adversarial sampling and type-constraint training technique (i.e., Local-cognitive w/o self-adv loss) contribute to performance improvement. The results also emphasize the effectiveness of our proposed local-cognitive negative sampling method, a non-trivial integration of the both above, in structured knowledge learning.

3.3.7 Analysis of Efficiency

Lastly, we discuss RatE's efficiency that is mainly brought by the following two factors. On the one hand, in line with previous trans-based graph embedding approaches, RatE only employs fast translation function and geometric distance measurement. On the other hand, even if a relation-adaptive translation function with weighted product is used in translating process, the function with few parameters has low time and space complexities. We compare RatE with a semantic matching graph embedding method TuckER (Balazevic et al., 2019) that uses a weight tensor to score a triple. As shown in Table 3.9, with competitive performance, TuckER requires much more learnable parameters than RatE for scoring. For example, TuckER has a weight tensor with $1,200,000$ parameters on WN18RR, whereas RatE only requires 88 parameters for all the eleven relations.

Chapter 4

Reasoning over Entity-Action-Location Graph for Procedural Text Understanding

4.1 Introduction

To effectively track the states and locations of entities, it is crucial to systematically model rich relations among various concepts in the paragraph, including entities, actions, and locations. Three types of relations are of particular interest.

First, mentions of the same entity in different sentences are related. The inherent relation among these mentions may provide clues for a model to generate consistent predictions about the entity. For example, the entity *electrical pulses* are mentioned in two sentences "*The retina's rods and cones convert it to electrical pulses. The optic nerve carries electrical pulses through the optic canal.*". Connecting its two mentions in two sentences helps to infer its location in the first sentence using the second sentence's information.

Second, detecting connections between an entity and the corresponding actions helps to make state predictions more accurate. Take the sentence "*As the encased bones decay, minerals seep in replacing the organic material.*" as an example. The entity *bone* is related to *decay* which indicates the state *destroy*, while it is not connected to *seep* indicating the state *move*. Given the relation between *bone* and *decay*, it is easier for the model to predict the state of *bone* as *destroy*, instead of being misled by the action *seep*.

Last, when the state or location of one entity changes, it may impact all associated entities. For example, in sentence "*trashbags are thrown into trashcans.*", *trash-*

bags are associated with *trashcans.* Then, in the following sentence "*The trashcan is emptied by a large trash truck.*", although *trashbags* are not explicitly mentioned, their locations are changed by the association with *trashcan.*

Recent works on procedural text understanding have achieved remarkable progress (Bosselut et al., 2018; Gupta and Durrett, 2019; Du et al., 2019; Das et al., 2019). However, the existing methods do not systematically model the relations among entities, actions, and locations. Instead, most methods either leverage inherent constraints on entity states or exploit external knowledge to make predictions. For example, Gupta and Durrett (2019) propose a structural neural network to track each entity's hidden state and summarize the global state transitions with a CRF model. Tandon et al. (2018) inject commonsense knowledge into a neural model with soft and hard constraints. Although Das et al. (2019) model the relation between entities and locations, there is no general framework to model the relations, and some important relations, such as entity-action and entity-entity relations, are ignored.

A general framework to systematically model the rich types of relations among entities, actions, and locations is essential to procedural text understanding. To the best of our knowledge, we are the first to explore comprehensive relation modeling, representation, and reasoning systematically. Specifically, we first construct an entity-action-location graph from a given paragraph, where three types of concepts (i.e., entities, locations, and actions) are identified and extracted as nodes. We then detect critical connections among those concepts and represent them as edges. Finally, we adopt a graph attention network to conduct **R**easoning over the **E**ntity-**A**ction-**L**ocation graph (REAL) (Huang et al., 2021), which provides expressive representations for downstream state and location predictions.

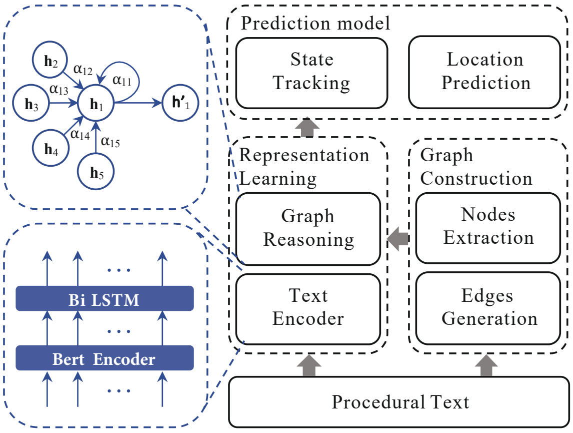


Figure 4.1 : An overview of REAL.

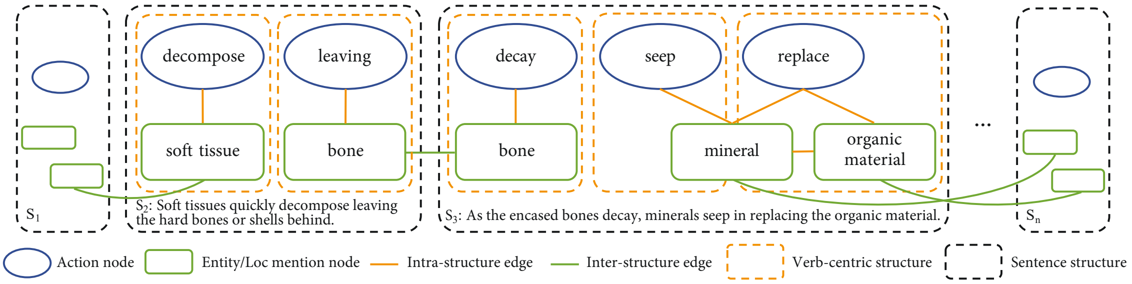


Figure 4.2 : An example of entity-action-location graph, constructed for paragraph "*...Soft tissues quickly decompose leaving the hard bones or shells behind. As the encased bones decay, minerals seep in replacing the organic material...* "

4.2 Approach

4.2.1 Task Definition.

The procedural text understanding task is defined as follows. Given a paragraph P consists of T sentences $(S_1, S_2, ..., S_T)$, describing the process (e.g., photosynthesis, erosion) of a set of N pre-specified entities $\{e_1, e_2, ..., e_N\}$, we need to predict the state y_t^s and location y_t^l for each entity at each step t corresponding to sentence S_t*. Candidate states are pre-defined (e.g., $y_t^s \in$ {not_exist (O), exist (E), move (M),

*We will use *step* and *sentence* interchangeably.

create (C), destroy (D)} in the ProPara dataset), and location y_t^l is usually a text span in the paragraph. Gold annotations for state and location at each step t are denoted as $\widetilde{y}_t^s$ and $\widetilde{y}_t^s$, respectively.

Figure 4.1 shows the overview of our approach, which consists of three main components: graph construction, graph-based representation learning, and prediction module. The graph construction module extracts nodes and edges from the input procedural paragraph and constructs a graph. The graph reasoning module initializes nodes representations using contextual word representations and reasons over the built graph. Finally, the prediction module leverages the graph-based representations to predict the state and location.

4.2.2 Graph Construction

Figure 4.2 shows an example of the graph constructed for a paragraph which describes *how fossil forms.* A semantic graph is denoted as $G = (N, E)$, where $N = \{n_i\}_{i=1}^K$ denotes all the nodes, and $E = \{e_i\}_{i=1}^L$ denotes all the edges.

Nodes Extraction. We first extract text spans as nodes from the given paragraph. The text spans in the extracted nodes should cover all essential concepts in the paragraph. Three types of concepts play an important role in the entity tracking task, i.e., actions, entity mentions, and location mentions. Therefore, we extract nodes for them and get all the nodes $N = \{N_a, N_e, N_l\}$ where N_a represents action nodes, N_e represents entity mention nodes, and N_l represents location mention nodes.

We first tag all the verbs by an off-the-shelf part-of-speech (POS) tagger[†] and construct a set of action nodes N_a with each node associated with a single verb or a phrase consisting of two consecutive verbs. For the entity mentions, we extract the

[†] https://github.com/flairNLP/flair

explicit (exact matching or matching after lemmatization) or implicit (pronouns) mentions of all the entities. Coreference resolution is used to find pronoun mentions in data pre-processing. Besides, we utilize the POS tagger to extract location mentions. Each tagged noun or consecutive phrase of adjective + noun is identified as a location mention.

Edges Generation. Capturing the semantic relations between various nodes is critical for understanding the process dynamics in the procedural text. To this end, we first derive verb-centric semantic structures via semantic role labeling (SRL)[‡] (Shi and Lin, 2019) for each sentence and then establish intra- and inter-semantic structure edges.

Given a verb-centric structure consisting of a central verb and corresponding arguments, we create two types of edges. (1) If an entity mention $n_e \in N_e$ or location mention $n_l \in N_l$ is a sub-string of an argument for verb $n_a \in N_a$, then we connect n_e/n_l to n_a. For example, for the sentence "*As the encased bones decay, minerals seep in replacing ...*", the verb *decay* has an argument *the encased bones* where *bones* is an entity mention, then we will connect the action node *decay* and entity mention node *bones.* (2) Two mentions in two arguments of the same verb are connected too. For example, for the sentence "*The trashbags are thrown into a large outdoor trashcan*", the verb *thrown* has two arguments, *the trashbags* and *into a large outdoor trashcan*, then we connect the two mention nodes *trashbags* and *trashcans.*

We also create edges between mentions of the same entity in different semantic structures. For example, in Figure 4.2, the entity *bones* are mentioned in two sentences, which correspond to two entity mention nodes. We connect these two nodes to propagate information from one to the other during graph-based reasoning.

[‡] https://github.com/allenai/allennlp.

4.2.3 Graph-based Representation Learning

Nodes Representation. We first feed the entire paragraph to the BERT (Devlin et al., 2019) model, which is then sent into a Bidirectional LSTM (Hochreiter and Schmidhuber, 1997) (BiLSTM) to obtain the contextual embedding for each token. Each node in our graph is associated with a text span in the paragraph. Therefore, the initial node representation is derived by mean pooling over all token embeddings in its corresponding text span. The contextual representation of node $n_i \in N$ is denoted as $\mathbf{h}_i$ $(i = 1, \ldots, K)$ with $\mathbf{h}_i \in \mathbb{R}^d$.

Graph Reasoning. We leverage a graph attention network (GAT) (Velickovic et al., 2018) for reasoning over the built graph. The network performs masked attention over neighbor nodes (i.e., connected with an edge) instead of all the nodes in the graph. We apply a two-layer GAT, which means each node can aggregate information from their two-hop neighbor nodes (nodes that can be reached within two edges).

In each GAT layer, we first extract a set of neighbor nodes $\mathcal{N}_i$ for each node n_i. The attention coefficients between node n_i and its neighbour n_j can be computed through a shared attention mechanism,

$$e_{ij} = \mathbf{a}^T[\mathbf{W}\mathbf{h}_i \| \mathbf{W}\mathbf{h}_j], \tag{4.1}$$

where $\mathbf{a} \in \mathbb{R}^{2d}$ and $\mathbf{W} \in \mathbb{R}^{d \times d}$ are learnable parameters, and $\|$ is the concatenation operation. We apply a LeakyReLU activate function and normalize the attention coefficients,

$$\alpha_{ij} = \mathrm{softmax}_j \left(\mathrm{LeakyReLU}\left(e_{ij}\right)\right). \tag{4.2}$$

Then, we aggregate the information from the neighbor nodes with multi-head attention to enhance the stability and efficiency. The aggregated feature for n_i with a

K-head attention can be represented as

$$\mathbf{h}'_i = \Big\Vert_{k=1}^{K} \sigma\left(\sum_{n_j\in\mathcal{N}_i} \alpha_{ij}^{k}\mathbf{W}^{k}\mathbf{h}_j\right) \tag{4.3}$$

in the first layer, and

$$\mathbf{h}''_i = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{n_j\in\mathcal{N}_i} \alpha_{ij}^{\prime k}\mathbf{W}^{\prime k}\mathbf{h}'_j\right) \tag{4.4}$$

in the second layer, where $\Vert$ is the concatenation operation, σ is the sigmoid activate function, $\mathbf{W}^k \in \mathbb{R}^{d\times d}$ is learnable matrix for kth head in first layer, and $\mathbf{W}'^k \in \mathbb{R}^{Kd\times d}$ is learnable matrix for kth head in second layer. α_{ij}^k and $\alpha_{ij}^{\prime k}$ are calculated with the corresponding $\mathbf{W}^k$ and $\mathbf{W}'^k$, respectively.

4.2.4 Prediction Model

Inspired by NCET (Gupta and Durrett, 2019), we track the state and location separately, by a state tracking and a location prediction module. Each module takes the representations of concerned nodes as input and outputs the prediction (i.e., state or location of an entity) at each time step.

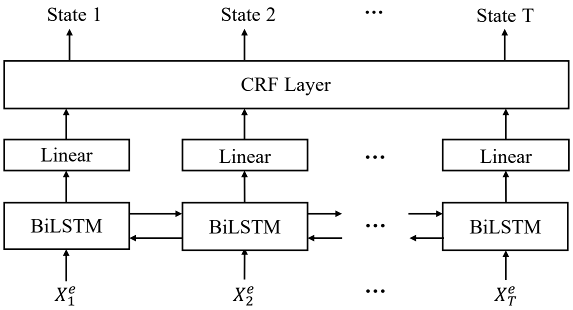


Figure 4.3 : Overview of state tracking model, which predicts states of the entity in every sentence S_t given entity e and paragraph P.

State Tracking. Given a paragraph P and an entity e, the state tracking module tracks the state of the entity for each sentence. We first generate the representations of all sentences for the entity. Considering that actions are good state-changing signals, we concatenate the embeddings of entity mention node and action node in the sentence as representation at step t. That is,

$$\mathbf{x}_t^e = \begin{cases} [\mathbf{h}_t^e \| \mathbf{h}_t^v], & \text{if } S_t \text{ contains } n_e \\ \mathbf{0}, & \text{otherwise} \end{cases} \tag{4.5}$$

where $\mathbf{x}_t^e$ denotes the representation of entity e in sentence S_t , $\mathbf{h}_t^e$ denotes the representation of the entity mention node n_e in sentence S_t, $\mathbf{h}_t^v$ denotes the representation of the action node n_a connected with n_e in sentence S_t. If entity e is not mentioned in sentence S_t, we use zero vector as representation of S_t for e. Note if there are multiple mention nodes for the entity e in sentence S_t, we take the mean pooling over all mention nodes as $\mathbf{h}_t^e$. And we take similar approach for multiple actions.

We utilize a BiLSTM layer on the sequence of sentence embeddings. And a conditional random field (CRF) (Durrett and Klein, 2015) is applied on the top of the BiLSTM to make the final prediction. The loss function for the state tracking module is defined as

$$L_{state} = -\sum_{(e,P)\in\mathcal{D}} \frac{1}{T}\sum_{t=1}^{T} \log\mathcal{P}\left(\widetilde{y}_t^s | P, e; \theta^G, \theta^{st}\right), \tag{4.6}$$

where $\mathcal{D}$ is the training collection containing entity-paragraph pairs, $\mathcal{P}\left(\widetilde{y}_t^s | P, e; \theta^G, \theta^{st}\right)$ represents the predicted probability of gold state $\widetilde{y}_t^s$ in sentence S_t given the entity e and paragraph P, θ^G are parameters for graph reasoning and the text encoder, and θ^{st} are parameters in state tracking module.

Location Prediction. For the location prediction module, we first collect all the location mention nodes as location candidates set $\mathcal{C}$. We add an isolated location

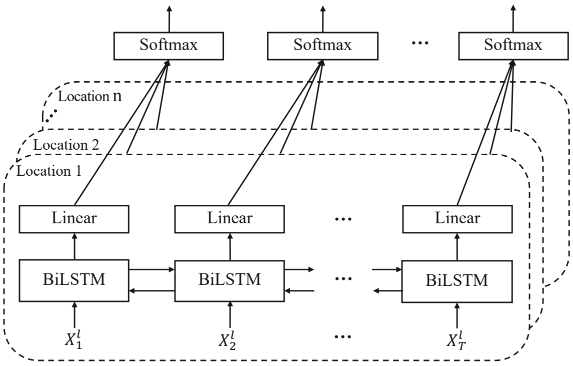


Figure 4.4 : Overview of location prediction model, which predicts locations of the entity in every sentence S_t given entity e and paragraph P.

node to represent the special location candidate '?', which means the location cannot be found in the paragraph. The representation of this node is randomly initialized and learnable during the training process.

Given an entity e and location candidate $l \in \mathcal{C}$, we represent the sentence S_t as

$$\mathbf{x}_t^l = [\mathbf{h}_t^e \| \mathbf{h}_t^l], \tag{4.7}$$

where $\mathbf{h}_t^e$ and $\mathbf{h}_t^l$ denotes the representation of the entity mention node and location mention node in sentence S_t. If the entity or location candidate is not mentioned in sentence S_t, we use a zero vector replacing $\mathbf{h}_t^e$ or $\mathbf{h}_t^l$.

We use a BiLSTM followed by a linear layer for the location predictor. The model outputs a score for each candidate at each step t. Then, we apply a softmax layer over all the location candidates' scores at the same step, resulting in a normalized probabilistic distribution. The location loss is defined as

$$L_{loc} = - \sum_{(e,P) \in \mathcal{D}} \frac{1}{T} \sum_{t=1}^{T} \log \mathcal{P} \left(\widetilde{y}_t^l | P, e; \theta^G, \theta^{loc} \right), \tag{4.8}$$

where $\mathcal{P} \left(\widetilde{y}_t^l | P, e; \theta^G, \theta^{loc} \right)$ represents the predicted probability of gold location $\widetilde{y}_t^l$ for entity e in sentence S_t, and θ^{loc} are parameters for location prediction module.

4.2.5 Learning and Inference

We create a single graph for each paragraph, which stays unchanged once created. Then the graph reasoning module and state/location prediction module are jointly trained in an end-to-end manner. The overall loss is defined as

$$L_{total} = L_{state} + \lambda_{loc} L_{loc}, \tag{4.9}$$

where λ_{loc} is the hyper-parameter to balance the state tracking and the location prediction loss.

We perform inference in pipeline mode. Specifically, for each entity, we first apply the state tracking module to infer its state at each time step. Then we only predict its location at steps when its state is changed (i.e., the predicted state is *create* or *move*[§]). And the locations of an entity with unchanged states can be inferred according to its locations in previous steps. Such pipeline fashion can increase consistency between states and locations of an entity than inferring location and state simultaneously.

4.3 Experiment

This section describes the evaluation results of REAL on two datasets (ProPara (Dalvi et al., 2018) and Recipes (Bosselut et al., 2018)). We also provide ablation study and case analysis to illustrate the effectiveness of graph-based reasoning.

4.3.1 Datasets and Evaluation Metrics

ProPara contains procedural texts about scientific processes, e.g., photosynthesis, fossil formulation. It contains about 1.9k instances (one entity-paragraph pair as

[§] The location of an entity will be *None* if its state is *destroy*. Therefore, we do not need to predict its location when an entity is *destroyed*.

Statistics	**ProPara**	**Recipes**
#sentences	3.3K	7.6K
#para	488	866
#train/#dev/#test	391/43/54	693/86/87
avg. #entities per para	4.17	8.57
avg. #sentences per para	6.7	8.8

Table 4.1 : Statistics of ProPara and Recipes dataset.

an instance) written and annotated by human crowd workers. We follow the official split (Dalvi et al., 2018) for train/dev/test set. The Recipes dataset consists of paragraphs describing cooking procedures and their ingredients as entities. We only use the human-labeled data in our experiment, with 80%/10%/10% of the data for train/dev/test, respectively. Detail statistics for the two datasets can be found in Table 4.1.

We follow previous work's setting (Dalvi et al., 2018) and evaluate the proposed approach on two types of tasks on the ProPara dataset, document-level task and sentence-level task. Document-level task focuses on figuring out input entities, output entities, entity conversions, and entity movements by answering corresponding questions. More details can be found in the official script[¶]. Following the official script, we evaluate models with averaged precision, recall, and F1 scores. In sentence-level task, we need to answer three categories of questions: (Cat-1) Is entity e created (destroyed, moved) in the process? (Cat-2) When is e created (destroyed, moved)? (Cat-3) Where is e created (destroyed, moved from/to)? For this task, we take macro-average and micro-average of the score for three sets of questions as

[¶] https://github.com/allenai/aristo-leaderboard/tree/master/propara evaluation metrics[‖].

For the Recipes dataset, we take the same setting as (Zhang et al., 2020), where the goal is to predict the ingredients' location changes during the process. We take precision, recall, and F1 scores to evaluate models[**].

4.3.2 Implementation Details

We use Bert base (Devlin et al., 2019) as encoder and reason with 3-heads GAT. Batch size is set to 16, and embedding size is set to 256. The learning rate r, location loss coefficient λ_{loc} and dropout rate d are derived by grid searching with in 9 trials in $r \in \{2.5 \times 10^{-5}, 3 \times 10^{-5}, 3.5 \times 10^{-5}\}$, $\lambda_{loc} \in \{0.2, 0.3, 0.4\}$, and $d \in \{0.3, 0.4, 0.5\}$. The implementation is based on Python and trained on a Tesla P40 GPU with Adam optimizer for approximately one hour (with approximately 112M parameters). We choose the best model with highest prediction accuracy on development set.

4.3.3 Main Results

Table 4.2 compares Real with previous work on the ProPara data for both document-level and sentence-level tasks. Our proposed approach consistently outperforms all previous models, which do not utilize external knowledge on all metrics. In particular, compared to DYNAPRO, it increases the document-level F1 score by 5.3%, and sentence-level macro averaged accuracy from 55.4% to 58.2%. Without any external data, our approach achieves comparable results to KOALA, which extensively leverages rich external knowledge in ConceptNet and Wikipedia pages, demonstrating the effectiveness of exploiting the entity-action-location graph. We also compare Real with the re-implemented NCET[††] on the Recipes dataset. As

[‖] https://github.com/allenai/propara/tree/master/propara/evaluation

[**] https://github.com/ytyz1307zzh/Recipes

[††] The re-implemented NCET achieves comparable accuracy with the previous state-of-the-art algorithm, DYNAPRO, i.e., 65.2% F1 score for NCET v.s. 65.5% for DYNAPRO.

Models	**Document-level task**			**Sentence-level task**				
	Precsion	**Recall**	**F1**	**Cat-1**	**Cat-2**	**Cat-3**	**Macro-Avg**	**Micro-Avg**
EntNet (Henaff et al., 2017)	54.7	30.7	39.4	51.6	18.8	7.8	26.1	26.0
QRN (Seo et al., 2017)	60.9	31.1	41.4	52.4	15.5	10.9	26.3	26.5
ProLocal (Dalvi et al., 2018)	81.7	36.8	50.7	62.7	30.5	10.4	34.5	34.0
ProGlobal (Dalvi et al., 2018)	48.8	61.7	51.9	63.0	36.4	35.9	45.1	45.4
ProStruct (Tandon et al., 2018)	74.3	43.0	54.5	-	-	-	-	-
XPAD (Dalvi et al., 2019)	70.5	45.3	55.2	-	-	-	-	-
KG-MRC (Das et al., 2019)	69.3	49.3	57.6	62.9	40.0	38.2	47.0	46.6
NCET (Gupta and Durrett, 2019)	67.1	58.5	62.5	73.7	47.1	41.0	53.9	54.0
DYNAPRO (Amini et al., 2020)	75.2	58.0	65.5	72.4	49.3	**44.5**	55.4	55.5
KOALA (Zhang et al., 2020)	77.7	**64.4**	70.4	**78.5**	53.3	41.3	57.7	57.5
REAL (our approach)	**81.9**	61.9	**70.5**	78.4	**53.7**	42.4	**58.2**	**57.9**

Table 4.2 : Experiment results on ProPara document-level task and sentence-level task. KOALA uses rich external data from Wikipedia and ConceptNet. Our approach achieves comparable performance to KOALA without any external knowledge.

shown in 4.3, REAL also surpass the strong baseline by 3.2%. All these results verify the effectiveness of the proposed graph-based reasoning approach.

4.3.4 Ablations

We conduct an ablation study to testify the effectiveness of multiple components in our approach. Table 4.4 and Table 4.3 list the results on ProPara and Recipes, respectively. As shown in Table 4.4, removing the graph-based representation learning for location/state prediction decreases the F1 score by 3.1%/3.6%, the gap becomes 4.4% without any graph-based reasoning. We can get similar observations on the Recipes dataset, indicating that exploiting the paragraph's rich relations is critical for both state tracking and location prediction.

Models	**Precsion**	**Recall**	**F1**
NCET re-implementation	**56.5**	46.4	50.9
Real	55.2	**52.9**	**54.1**
-Location	54.9	51.7	53.3
-State	54.9	52.0	53.4
-Graph	57.2	47.9	52.1

Table 4.3 : Comparison on Recipes dataset.

Models	**Precsion**	**Recall**	**F1**
Real	81.9	61.9	70.5
-Location	81.0 (-0.9)	57.7 (-4.2)	67.4 (-3.1)
-State	73.7 (-8.2)	61.2 (-0.7)	66.9 (-3.6)
-Graph	72.0 (-9.9)	61.2 (-0.7)	66.1 (-4.4)

Table 4.4 : Ablation study on ProPara dataset.

4.3.5 Analyses of Different Relations

To further illustrate the effectiveness of different types of relations, we conduct below analyses and present three cases with predictions of Real with and without graph reasoning in Figure 4.5.

First, to verify the effectiveness of action-entity relations in multi-verb sentences, we compare Real of with and without graph reasoning on sentences containing multiple (i.e., more than 2) verbs in Table 4.5. We figure out that graph-based reasoning increases the performance by 5.7%, indicating that accurately connecting entities and corresponding actions improves the prediction accuracy. For case 1 shown in Figure 4.5, the relation between the entity *bone* the action *decay* helps the model to correctly predict the state of *bone* as *destroy* since the action *decay*

Segments	Models	Precision	Recall	F1
muli-verb	w/o graph	73.0	58.2	64.8
	w/ graph	**82.5**	**61.0**	**70.1**
implicit	w/o graph	74.9	57.9	65.3
	w/ graph	**83.7**	**60.3**	**70.1**

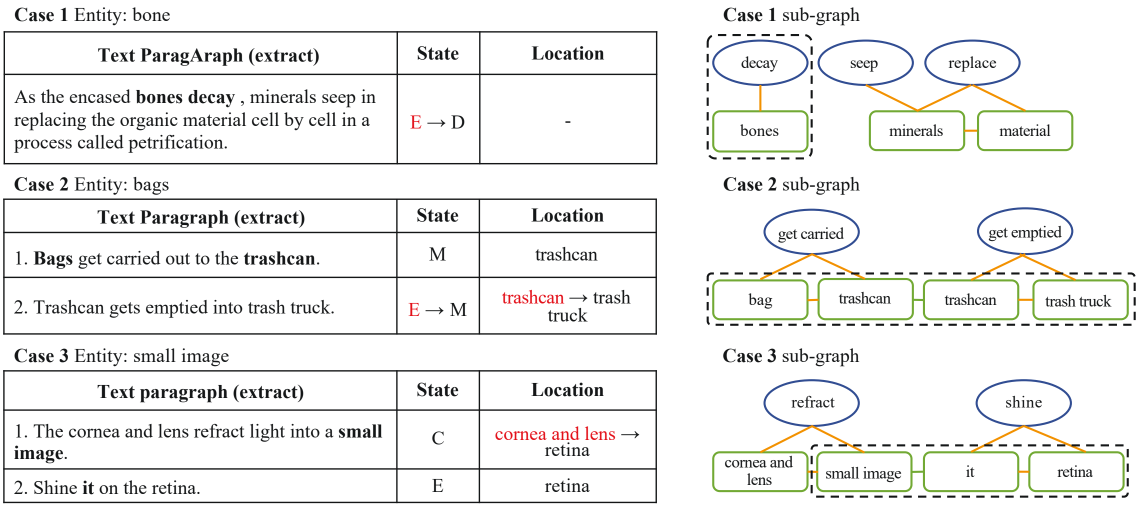
Table 4.5 : Analyses of impact of entity-action and entity-entity relations on ProPara.

indicates *destroy*. However, without such accurate connection between *bone* and *decay*, the prediction model is very likely to be misled by other actions such as *seep* or *replace*.

Second, we illustrate the impact of entity-entity relations by comparing our approach and baseline where the entity is not explicitly mentioned[‡‡]. As shown in Table 4.5, Real increase the accuracy by 4.8%, which indicates the effectiveness of our approach by modeling cross-entity relations. The second case in Figure 4.5 illustrates the effectiveness of using entity-entity relations. The entity *bags* is not explicitly mentioned in the sentence "*Trashcan gets emptied into trash truck*", and thus the baseline model cannot correctly predict its state and location. However, connecting it to the entity *trashcan* which is derived in the first sentence, helps the model infer its state and location correctly.

Third, as discussed in section 4.1, mention-mention connections might improve accuracy when there are multiple mentions for the same entity. The third case in Figure 4.5 shows how Real utilizes relations between different mentions for the same entity. In the first sentence, the location of entity *small image* is not mentioned,

[‡‡] We only compare performance for those entity-sentence pairs with gold state as *Move*, *Create* and *Destroy*.



Case 1 Entity: bone

Text ParagAraph (extract)	State	Location
As the encased **bones decay** , minerals seep in replacing the organic material cell by cell in a process called petrification.	E → D	-

Case 2 Entity: bags

Text Paragraph (extract)	State	Location
1. **Bags** get carried out to the **trashcan**.	M	trashcan
2. Trashcan gets emptied into trash truck.	E → M	trashcan → trash truck

Case 3 Entity: small image

Text paragraph (extract)	State	Location
1. The cornea and lens refract light into a **small image**.	C	cornea and lens → retina
2. Shine **it** on the retina.	E	retina

Figure 4.5 : Examples of model predictions of our approach w/ (black) and w/o (red) graph reasoning. Corresponding sub-graph is plot on the right of the paragraph. Dotted rectangles in the sub-graph highlight key connections for correct prediction in graph-based reasoning.

which results in wrong location prediction when no graph reasoning is used. In contrast, the built graph connects this mention with preposition *it* in the second sentence where its location is revealed as *retina*. Therefore, our model correctly predicts *small image*'s location by graph-based representation learning.

4.3.6 Error Analyses

We randomly sample 100 wrongly predicted examples and summarize them into the following types.

First, the ambiguity between similar entities makes it difficult to derive accurate representations for them. For instance, *fixed nitrogen* and *gas-based nitrogen* are two different entities related to nitrogen in the paragraph "*Nitrogen exists naturally in the atmosphere. Bacteria in soil fix the nitrogen. Nitrogen is now usable by living things.*". It is difficult for a model to distinguish which entity the mention *nitrogen* refers to.

Second, commonsense knowledge is required. For example, it is difficult to infer the location of the entity *bone* in the sentence "*An animal dies. It is buried in a watery environment.*" without the knowledge "*bone is part of animal*". Therefore, injecting appropriate external knowledge while avoiding noise may improve the model.

Third, similar actions indicate different states in different contexts. For instance, in sentence "*the tree eventually dies.*", the state of *tree* is labeled as *destroy*, while in sentence "*most fossils formed when animals or plants die in wet environment.*", the state of *animals* and *plants* are all annotated as *exist*, which may confuse the model.

Chapter 5

Improve Temporal Reading Comprehension via Precise Question Understanding

5.1 Introduction

A natural solution for temporal ordering understanding is to compare each candidate answer and the referred event in the question and classify their temporal relation into several pre-defined categories, e.g., UzZaman et al. (2013) defines 13 possible relations such as *after, ends, equal to.* Nonetheless, since temporal relationships vary greatly, it is almost impossible to enumerate all possible relationships. Figure 4.5 shows several divergent varieties of temporal relations: one might query about *plain after* in Q1, *negated after* in Q2, *constrained after* in Q3, etc. Similarly, a question might query about *usually happen* in Q4, *might happen*, or other relations. Moreover, creating sufficient labels for all such relations is costly and poses great challenges for real-world applications. Therefore, the classification-based approach is incompetent to handle the flexible relations in temporal reading comprehension.

Another paradigm is to formulate it as a reading comprehension problem and directly predict the answer to a question. With the help of large pre-trained language models (e.g., BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019)), such approaches have achieved relatively good performance. However, they still struggle for the temporal reading comprehension task due to the lack of precise question understanding. For example, given the same passage, the BERT model fine-tuned on SQuAD (Rajpurkar et al., 2016) predicts the same answer to the two questions (Ning et al., 2020), "*What happened before a woman was trapped*" and "*What happened* *after a woman was trapped*". In this case, although the two questions share almost the same words, the only different one between *before* and *after* leads to completely opposite intentions. Moreover, even if two questions query about similar relations, different varieties might also lead to various answers. Take the question Q4 "*What usually happened during the press release?*" and "*What might happen during the press release?*" in Figure 4.5 as an example. Although they both query about events occurring after *the press release*, the slight difference conveys divergent semantics and leads to different answers.

To tackle these challenges, we propose a novel question answering approach with precise question understanding (Huang et al., 2022). Intuitively, temporal ordering questions consist of two elements, referred events, and concerned temporal relations. For example, the question "*What usually happened during the press release?*" can be decomposed into the referred event *the press release* and the concerned relation *usually happen during*. Inspired by this observation, we first encode such questions into two representations, the event vector $\mathbf{h_c}$ and the relation vector $\mathbf{h_r}$. Then we evaluate how well each candidate answer matches the relation $\mathbf{h_r}$ compared to $\mathbf{h_c}$ with a separate MLP module. Such fine-grained representations enable a better understanding of questions by focusing on different elements with different vectors and further provides good interpretability about the reasoning process. More importantly, it empowers the model to capture the semantics of divergent variants of temporal relations. Specifically, we harness an auxiliary contrastive loss that aims to distinguish relations with subtle but critical changes.

5.2 Approach

We first introduce the definition of temporal reading comprehension (TRC) and then describe the model architecture consisting of contextual encoder, question understanding, and event relation assessment. Finally, we provide details for the learning and inference process.

5.2.1 Task Definition

The Temporal Reading Comprehension (TRC) task is defined as follows. Given a passage P which describes a set of events, a system is required to answer a temporal ordering question Q. Here *events* refer to verbs or nouns which define actions or states. A temporal ordering question usually queries events satisfying some concerned temporal relations considering one or more referred events. For example, the first passage in Figure 4.5 describes events about *Hamas goverment*, and question $Q1$ queries which events have the temporal relation *happen after* with the referred event *the victory*. The answer set A to a question Q could be empty when no events meet the requirement.

5.2.2 Model Architecture

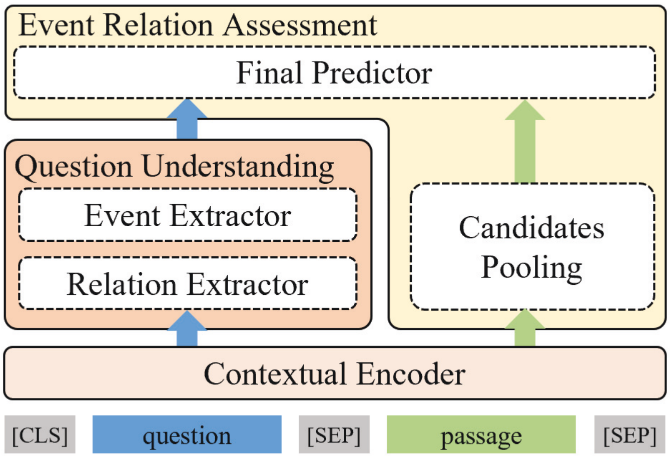


Figure 5.1 : An overview of the proposed model.

Figure 5.1 depicts the proposed model architecture. Specifically, the passage P and question Q are first encoded by a *contextual-aware encoder*, after which the representations of the question are passed to a *question understanding* module. Finally, each candidate answer is evaluated considering whether it satisfies the concerned relation to the referred event by an *event relation assessment* module.

Contextual Encoder We first encode the passage-question pairs with a pre-trained language model encoder, and here we take BERT as an example. Given a question $Q = [q_i]_{i=1}^{m}$ and a passage $P = [p_i]_{i=1}^{n}$, where m and n are token numbers, we concatenate them into a sequence with the format of *[cls] question [sep] passage [sep]*, which is then fed into the contextual encoder to generate the embeddings,

$$[\mathbf{h_1^q}, ..., \mathbf{h_m^q}, \mathbf{h_1^p}, ..., \mathbf{h_n^p}] = \text{BERT}([q_1, ..., q_m, p_1, ..., p_n]), \tag{5.1}$$

where $\mathbf{h_i^q}, \mathbf{h_i^p} \in \mathcal{R}^d$ are embeddings for question token q_i and passage token p_i, and d is the embedding size.

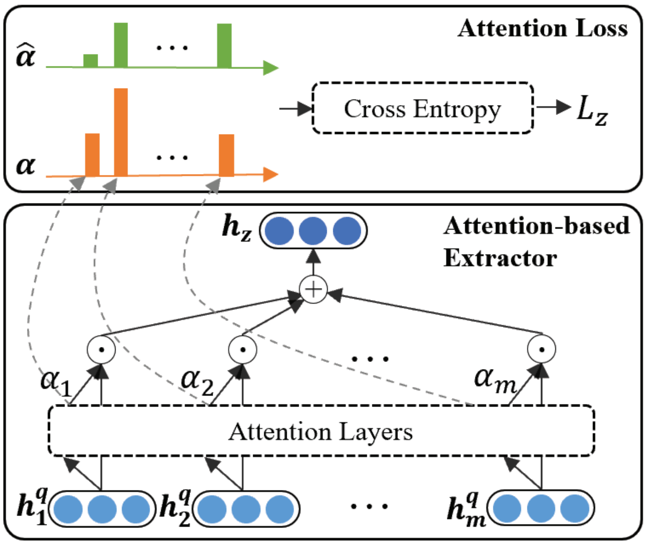


Figure 5.2 : The structure of attention-based event/relation extractor, with attention loss for it.

Question Understanding As discussed in Section 5.1, precise question understanding plays an essential role in TRC task. Therefore, we propose a question understanding module to achieve that. Intuitively, a temporal ordering question consists of two elements, referred events, and concerned temporal relation. For example, the question "*What usually happened during the press release*" queries the temporal relation *usually happen* to the event *the press release.* A straightforward solution is to decompose the question into two segments directly. However, natural language questions vary a lot, and hard decomposition is risky and might propagate errors to downstream modules, which is verified by experimental analysis in Section 5.3.5,

Therefore, we design an attention-based extractor to decompose the question implicitly, and obtain two hidden representations, $\mathbf{h_c}$ for the referred event and $\mathbf{h_r}$ for the concerned temporal relation as follows,

$$\mathbf{s_i^{(z)}} = \tanh(\mathbf{W_1^{(z)}} \mathbf{h_i^q} + b_1^{(z)}), \quad z \in \{c, r\} \tag{5.2}$$

$$\alpha_i^{(z)} = \text{softmax}(\mathbf{W_2^{(z)}} \mathbf{s_i^{(z)}} + b_2^{(z)}), \quad z \in \{c, r\} \tag{5.3}$$

$$\mathbf{h_z} = \sum_{i=1}^{m} \alpha_i^{(z)} \mathbf{h_i^q}, \quad z \in \{c, r\} \tag{5.4}$$

where $\mathbf{W^{(c)}}, \mathbf{W^{(r)}} \in \mathcal{R}^d$, and $b^{(c)}, b^{(r)} \in \mathcal{R}$ are learn-able weights for the extractor, $\mathbf{h_i^q} \in \mathcal{R}^d$ is the embedding for the i-th question token. To effectively learn $\mathbf{h_r}$ and $\mathbf{h_c}$, we employ several auxiliary losses in the training phase, which will be described in section 5.2.3.

Event Relation Assessment Given the question representations $\mathbf{h_r}$ and $\mathbf{h_c}$, the *event relation assessment* module evaluates how a candidate answer satisfy the relation $\mathbf{h_r}$ with respect to $\mathbf{h_c}$. Let $e = p_i \ldots p_{i+l}$ denotes the candidate answer, which consists of l tokens in the passage P. We first get the representation of e by pooling

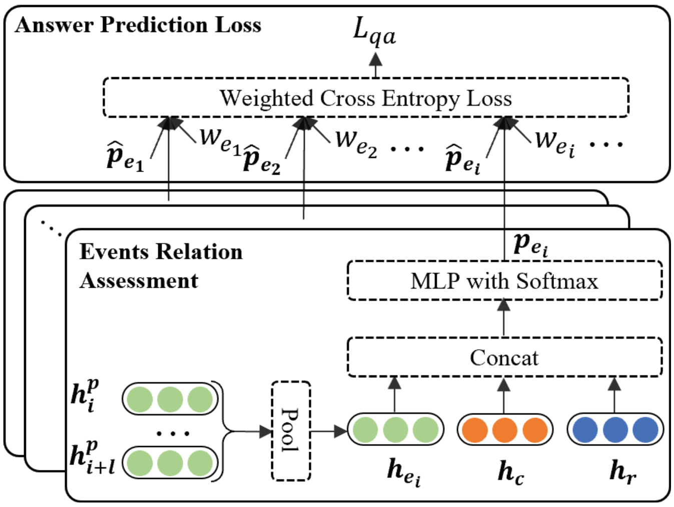


Figure 5.3 : The structure of the event relation assessment, with answer prediction loss for it.

over according token vectors,

$$\mathbf{h_e} = \text{Pool}(\mathbf{h_i^p}, \ldots, \mathbf{h_{i+l}^p}). \tag{5.5}$$

Then we concatenate the representations of the candidate event $\mathbf{h_e}$, question relation $\mathbf{h_r}$, and the question event $\mathbf{h_c}$, and feed it into a two-layer MLP, followed by a softmax function to get the final probability,

$$\mathbf{o_e} = \tanh(\mathbf{W_1^o}[\mathbf{h_e}; \mathbf{h_c}; \mathbf{h_r}] + \mathbf{b_1^o}), \tag{5.6}$$

$$\mathbf{p_e} = \text{softmax}(\mathbf{W_2^o}\mathbf{o_e} + \mathbf{b_2^o}), \tag{5.7}$$

where $\mathbf{W_1^o} \in \mathcal{R}^{3d \times d'}$, $\mathbf{W_2^o} \in \mathcal{R}^{d' \times 2}$, $\mathbf{b_1^o} \in \mathcal{R}^{d'}$, $\mathbf{b_2^o} \in \mathcal{R}^{2}$ are model parameters, and ; indicates concatenation. $\mathbf{p_e} \in \mathcal{R}^2$ is the probability whether the candidate e satisfies the temporal relation h_r with respect to event h_c.

5.2.3 Learning Objectives

We employ three learning objectives for model training, including a classification loss L_{qa} function for final answer prediction, and an attention loss L_{att} and a contrastive loss L_{con} for precise question understanding. The overall loss is a weighted combination of all the objectives,

$$\mathcal{L} = w_{qa} L_{qa} + w_{att} L_{att} + w_{con} L_{con}. \tag{5.8}$$

Answer Prediction Loss The training objective for final answer prediction is defined as,

$$L_{qa} = -\sum_{e \in \mathcal{C}} w_e \hat{\mathbf{p}}_{\mathbf{e}}^T \log \mathbf{p}_{\mathbf{e}}, \tag{5.9}$$

where $\mathcal{C}$ is the candidate event set, w_e is the weight for candidate e, $\mathbf{p}_{\mathbf{e}} \in \mathcal{R}^2$ is the predicted probability from Eq. (5.7), and $\hat{\mathbf{p}}_{\mathbf{e}} \in \{0, 1\}^2$ is the golden label indicating whether the candidate e belongs to the final answer of the question.

Usually, the candidate set $\mathcal{C}$ is derived by preliminary filtering all unigrams in the passage P. However, some candidates are easy to be classified while others are not. For example, it is easy to classify the word *government* in Figure 4.5 as a negative answer since it is not an event. In contrast, predicting whether the word *frozen* is the answer for Q1 in Figure 4.5 is more challenging. Inspired by this observation, we assign weights w_e for candidates in the learning objective, $w_e = 1.5$ if e is an event, and otherwise $w_e = 1.0$. The label of whether a word is an event can be derived when labeling the final answer with little effort, so we can safely assume that we always have such annotation[*].

[*] The TORQUE dataset in our experiment also contains such annotation, and we use it directly in our approach

Attention Loss Besides the answer prediction loss, we also leverage an auxiliary loss to guide the learning of the attention score $\alpha_i^{(c)}$ and $\alpha_i^{(r)}$ defined in Eq. (5.3). We first derive silver annotation for referred events and concerned relation in a passage using a rule-based approach, which will be detailed in Section 5.3.2. Let Q_c, Q_r be the set of event and relation tokens according to the silver annotation. Then we have $\hat{\alpha}_i^{(z)} (z \in \{c, r\})$ as the derived attention label,

$$\hat{\alpha}_i^{(z)} = \begin{cases} \frac{1}{|Q_z|}, & \text{if } q_i \in Q_z, \\ 0, & \text{otherwise.} \end{cases} \tag{5.10}$$

The attention loss is defined as,

$$L_{att} = L_c + L_r, \tag{5.11}$$

where

$$L_z = -\sum_i \hat{\alpha}_i^{(z)} \log \alpha_i^{(z)}, \quad z \in \{c, r\}. \tag{5.12}$$

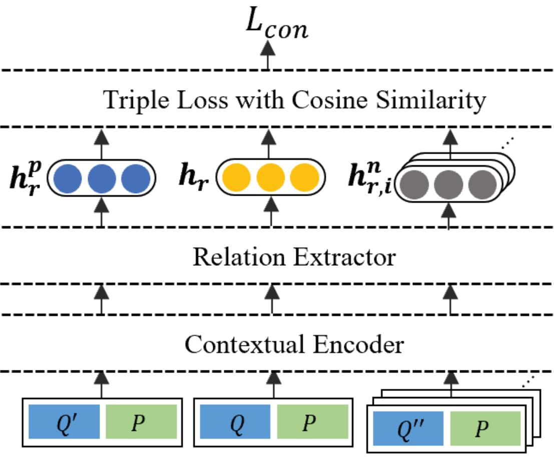


Figure 5.4 : Illustration of the contrastive loss for question understanding.

Contrastive Loss As shown in Figure 4.5, a small change of a question might lead to substantially divergent temporal relations. To this end, we propose to leverage a contrastive loss for precise learning of question relation representations.

For the relation representation $\mathbf{h_r}$ of a question Q, we derive a positive vector $\mathbf{h_r^p}$ and a set of negative ones $\{\mathbf{h_{r,i}^n}\}_{i=1}^{N}$). The positive sample $\mathbf{h_r^p}$ is obtained in two ways. First, we search questions with the same temporal relations but different events, from which we randomly sample one and take its relation representation as $\mathbf{h_r^p}$. Note we can get the silver annotation of events and relations in a question by a rule-based approach. Please refer to section 5.3.2 for more details. Second, if no such questions can be found, we take the similar approach as in SimCSE (Gao et al., 2021), which applies a different dropout on $\mathbf{h_r}$ and gets a variant of $\mathbf{h_r}$ as $\mathbf{h_r^p}$. We search questions that contain the same events by different temporal relations with respect to Q, and take their relation representations as the negative set $\{\mathbf{h_{r,i}^n}\}_{i=1}^{N}$).

Given the triple $(\mathbf{h_r}, \mathbf{h_r^p}, \{\mathbf{h_r^n}\})$ for the question Q, its loss is defined as,

$$L_{con}(Q) = -\log \frac{e^{\cos(\mathbf{h_r}, \mathbf{h_r^p})}}{e^{\cos(\mathbf{h_r}, \mathbf{h_r^p})} + \frac{1}{N}\sum_{i=1}^{N} e^{\cos(\mathbf{h_r}, \mathbf{h_{r,i}^n})}}, \tag{5.13}$$

where $cos()$ indicates cosine similarity.

5.2.4 Inference

The inference phase takes three steps. First, we generate a candidate set $\mathcal{C}_p$ for each passage P. Generally speaking, one can take any n-gram in P as a candidate. In temporal relation understanding, we usually take a triggering word as an event candidate. Therefore, $\mathcal{C}_p$ is the set of all unigrams in P. Then, we filter $\mathcal{C}_p$ according to part-of-speech (POS) tagging. Specifically, we use an off-the-shelf POS tagger to tag all words in P, and then keep only verbs and nouns in $\mathcal{C}_p$. Finally, each candidate $e \in \mathcal{C}_p$ together with the passage P and the question Q is fed into our proposed model, and e is evaluated according to Eq. (5.7) and gets its score $\mathbf{p_e}$, where $p_{e,0}$ represents the probability that the candidate matches the question Q. Then we can get the final answer set A as $A = \{e : e \in \mathcal{C}_p \text{ and } p_{e,0} > \tau\}$, where τ is a predefined threshold.

5.3 Experiment

This section describes an empirical evaluation of our proposed approach. We also provide analysis, ablation studies, and case analysis to demonstrate its effectiveness.

5.3.1 Settings

Dataset We evaluate the proposed approach on the TORQUE dataset. TORQUE is a temporal reading comprehension benchmark. Each training sample contains a passage and a question requiring understanding temporal relation between events in the passage. Figure 4.5 shows several examples of training data. The answer to a question consists of an event set A, and A could be empty if no event in the passage satisfies the requirement of the question. In TORQUE, events are defined as event triggers, usually verbs or nouns describing actions or states. There are 3.16k passages with 30.7k questions in total and 2 events for an answer on average. We follow the official split[†] with 80%/5%/15% of data in training/validation/test.

Evaluation Metrics Following (Ning et al., 2020)[‡], we report three metrics in our experiment, including standard macro F1 and Exact Match (EM) for question answering and consistency score(C). There are multiple annotations for each passage-question pair, which might not always be consistent with each other. We follow the official implementation. Specifically, for each sample, a model's prediction is evaluated according to all annotations, where the largest score is selected and aggregated as the final result.

[†] https://github.com/qiangning/TORQUE-dataset

[‡] https://github.com/rujunhan/TORQUE

5.3.2 Implementation Details

We experiment four pre-trained language models as our contextual encoder, i.e., the base and large model of BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019). The embedding size d is set to 64, d' in Eq (5.6) and Eq (5.7) is set to 64. The threshold τ for inference is set to 0.5. In model training, the batch size is set to 16, the dropout rate is set to 0.5. The combination weight w_{qa}, w_{att} and w_{con} in Eq. (5.8) is set to 1.0, 0.3, and 1.0, respectively. We search the learning rate lr, with grid searching within 3 trials in $lr \in \{0.9 \times 10^{-5}, 1.0 \times 10^{-5}, 1.1 \times 10^{-5}\}$ for the base and large model of RoBERTa, and $lr \in \{4.0 \times 10^{-5}, 5.0 \times 10^{-5}, 6.0 \times 10^{-5}\}$ for the base and large model of BERT. The implementation is based on Python and trained on a Tesla V100 GPU with Adam optimizer for approximately three hours (base model with approximately 110M parameters) and ten hours (large model with approximately 340M parameters). We get the averaged result of three trials for each setting, choose the model with the highest F1 score on the development set, and report the performance on the test set derived from the official online test[§].

Deriving Attention Annotation The relation annotation Q_r for question Q is derived as follows. First, we compile a dictionary for temporal relations, such as *before*, *after*, etc. Please refer to Appendix A.1 for the complete list. Then Q_r is constructed with those words in Q that hit the dictionary. The event annotation Q_c is mainly derived according to the passage P. Particularly, we assume the mentioned event list E in P is known. If a word of Q matches an event in E, it is included in Q_c. Otherwise, if no words of Q hit E, we rely on the relation annotation. Suppose the last relation word is in position k, then $Q_{k+1...n}$ is set as Q_c.

	Dev			Test		
	F1	EM	C	F1	EM	C
BERT-base						
baseline†	67.6	39.6	24.3	67.2	39.8	23.6
Ours	70.5	44.6	26.2	69.8	43.0	26.1
Δ	+2.9	+5.0	+1.9	+2.6	+3.2	+2.5
BERT-large						
Baseline†	72.8	46.0	30.7	71.9	45.9	29.1
Ours	73.5	46.5	31.8	72.6	45.1	30.1
Δ	+0.7	+0.5	+1.1	+0.7	-0.8	+1.0
RoBERTa-base						
Baseline†	72.2	44.5	28.7	72.6	45.7	29.9
Ours	73.3	47.0	32.5	73.5	46.8	31.5
Δ	+1.1	+3.5	+3.8	+0.9	+1.1	+1.6
RoBERTa-large						
Baseline†	75.7	50.4	36.0	75.2	**51.1**	34.5
Ours	**77.5**	**52.2**	**37.5**	**76.1**	51.0	**38.1**
Δ	+1.8	+1.8	+1.5	+0.9	-0.1	+3.6
Human	-	-	-	95.3	84.5	82.5

Table 5.1 : Comparison of our approach and the baseline on the TORQUE Dataset. † denotes published results (Ning et al., 2020).

5.3.3 Main Results

We compare our approach with the baseline (Ning et al., 2020), which takes a passage and the corresponding question as input and applies a one-layer perception on the embedding of each token to predict whether it is the answer of the question

§https://leaderboard.allenai.org/torque/submissions/public or not. The comparison results with four different contextual encoders are shown in Table 5.1. The table shows that our proposed approach outperforms the baseline on nearly all evaluation metrics. Our model achieves state-of-the-art results with the RoBERTa-large encoder, increasing the F1 score by 1.8% and 0.9% for the dev and test set, respectively. We can see a huge increase for the consistency score (C) on the test set from 34.5% to 38.1%. Using other pre-train language models like BERT-base, our model also improves the performance compared to the baseline approach, by 2.6%, 3.2%, 2.5% in terms of F1, EM, and C score, respectively. Although there is still a large gap towards the human performance, our model takes a large step compared to the baseline approach, verifying the effectiveness of the proposed approach.

5.3.4 Ablation Study

Models	**F1**	**EM**	**C**
Our Model	76.1	51.0	38.1
-con	75.8 (-0.3)	49.8 (-1.2)	37.0 (-1.1)
-con -att	75.6 (-0.5)	50.8 (-0.2)	36.6 (-1.5)
-w_e	75.8 (-0.3)	50.6 (-0.4)	37.6 (-0.5)
-all	74.8 (-1.3)	49.7 (-1.3)	34.0 (-4.1)

Table 5.2 : Ablation study on the test set of TORQUE. RoBERTa-large is used as contextual encoder.

We conduct an ablation study to illustrate the effectiveness of each loss in our approach. As shown in Table 5.2, removing the contrastive loss will lead to a 1.1% drop on consistency value. When we remove both the contrastive and attention loss for question understanding and use mean pooling over the contextual embedding of the whole question token sequence, the macro F1 score and the consistency score decrease by 0.5% and 1.5%, respectively, showing that precise question understanding plays a critical role for TRC. Also, we remove weight w_e in the answer prediction loss in Eq. (5.9), which results in a 0.3% drop in terms of the F1 score. When all auxiliary loss is removed, which is basically the same as the baseline model with our own implementation, it leads to a huge gap of 1.3%, 1.3%, 4.1% on macro F1, exactly match and Consistency score, respectively. The results of the ablation study indicate that each element of our proposed model is critical for temporal relation understanding.

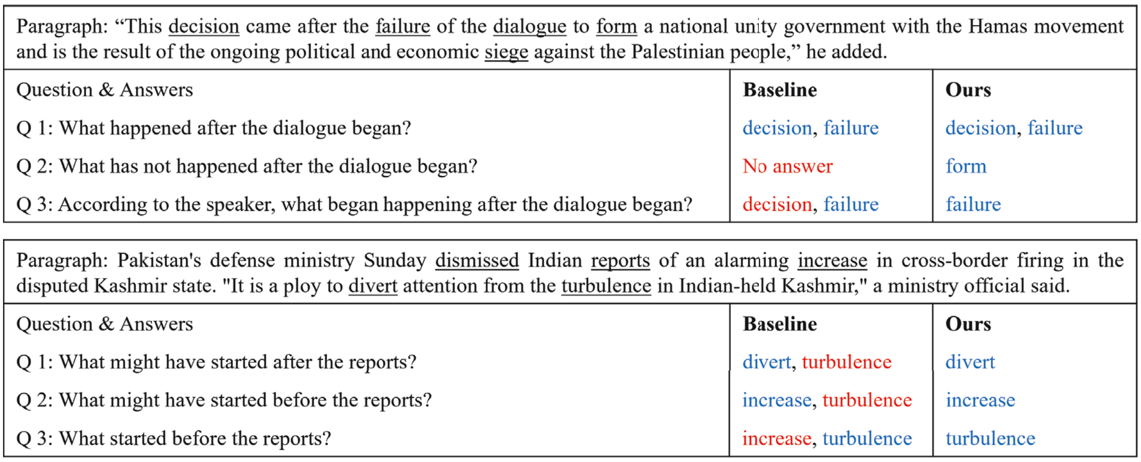
5.3.5 Question Representation Analysis

Models	**F1**	**EM**	**C**
w contrastive loss			
attention-based	76.1	51.0	38.1
rule-based	75.8 (-0.3)	50.6 (-0.4)	37.6 (-0.5)
w/o contrastive loss			
attention-based	75.8	49.8	37.0
rule-based	75.6 (-0.2)	48.9 (-0.9)	36.3 (-0.7)

Table 5.3 : Comparison of attention-based and rule-based question representation learning. RoBERTa-large is used as contextual encoder.

As discussed in Section 5.2.2, a straightforward solution for question understanding is to decompose a temporal ordering question into two parts directly. This section compares our attention-based approach with the hard question decomposition, which obtains the two question vectors $\mathbf{h_r}$ and $\mathbf{h_c}$ by conducting mean pooling over embeddings of tokens in Q_r and Q_c respectively. The comparison results are shown in Table 5.3. We can see that although the rule-based approach achieves relatively good accuracy, it still underperforms our attention-based approach. For example, when no contrastive loss is employed, the EM score drops by 0.9% when replacing the attention-based representation with the rule-based one. The possible reason is that the rule-based decomposition cannot handle all questions perfectly, and errors in the decomposition will be propagated to downstream modules. For example, "*What could have happened while the announcement was made but didn't?*". "*but didn't*" is a crucial negate in the temporal relation, but the rule-based method might miss it.

5.3.6 Case Study



Paragraph: "This decision came after the failure of the dialogue to form a national unity government with the Hamas movement and is the result of the ongoing political and economic siege against the Palestinian people," he added.

Question & Answers	**Baseline**	**Ours**
Q 1: What happened after the dialogue began?	decision, failure	decision, failure
Q 2: What has not happened after the dialogue began?	No answer	form
Q 3: According to the speaker, what began happening after the dialogue began?	decision, failure	failure

Paragraph: Pakistan's defense ministry Sunday dismissed Indian reports of an alarming increase in cross-border firing in the disputed Kashmir state. "It is a ploy to divert attention from the turbulence in Indian-held Kashmir," a ministry official said.

Question & Answers	**Baseline**	**Ours**
Q 1: What might have started after the reports?	divert, turbulence	divert
Q 2: What might have started before the reports?	increase, turbulence	increase
Q 3: What started before the reports?	increase, turbulence	turbulence

Figure 5.5 : Case study of our approach and the baseline model. Correct answers are marked in blue. Incorrect ones are marked in red. Candidate events in passages are underlined. Both the baseline and our approach use RoBERTa-large as encoder.

Figure 5.5 shows predicted answers of our model and the baseline for several questions. For the first passage, Questions 1, 2, and 3 inquire about the "*happened after*" temporal relation, but with subtle differences. Q1 is the most common form, which can be answered correctly by both the baseline and our proposed approach. Meanwhile, the baseline model can not capture the negation information in Q2 and fails to predict the correct answer. In Q3 "*happened after*" is constrained by the word *begin*, which confuses the baseline model and leads to partially correct answers. In contrast, our proposed approach can capture these subtle but critical differences and thus makes correct predictions.

For the second passage, our proposed model performs better for all three questions of different temporal types. Q1 and Q2 are variants of uncertain relations, which query about two opposite temporal relations "*started after*" and "*started before*". The word "*might*" brings uncertainty for the concerned temporal relation, which confuses the baseline model, leading to the wrong prediction for the candidate answer "*turbulence*" for both questions. Q3 queries about a popular temporal relation, and our model can precisely capture the difference between it and two other ones and predict that the candidate event "*increase*" does not meet its requirement since it comes from a controversial report.

5.3.7 Error Analysis

We randomly sample 100 wrongly predicted question-passage pairs from the validation set, which can be summarized into three categories.

Multi-round Reasoning Sometimes one needs to perform multi-round reasoning to infer the relation between two events, for example, given the passage "*Roughly 40 minutes after the operation began, jubilant soldiers appeared on the rooftop of the residence, flashing the V victory sign. Then Fujimori, who ordered the operation, arrived to tour the residence and embraced the freed hostages.*", the temporal ordering between "*ordered*" and "*the jubilant soldiers appeared on the rooftop*" is inferred by multi-step reasoning. That is, "*ordered*" happened before "*operation began*", and "*operation began*" happened before "*solder appeared*", and thus "*ordered*" happened before "*appeared*". An advanced reasoning framework is necessary to handle such cases, and we leave it as future work.

Commonsense Knowledge Required The given passage might not provide sufficient information. For example, in the passage "*He was preparing the paperwork for the move, following the course of an absolutely standard transfer. Sadly he killed himself at home in the meantime.*", although it states that "*preparing the paperwork*" and ""*he killed himself*" happened "*in the meantime*", commonsense knowledge indicates that *one cannot kill himself and prepare the paperwork at the same time.* So we can infer that "*preparing*" happened before "*killed*". Incorporating external knowledge is a potential solution for such cases.

Ambiguous Labeling Since the concept of event is not well-defined, it might lead to ambiguous labeling. Considering a passage contains a span "*decision is made*", some annotators might label *decision* as a candidate event, while others does not. This causes inconsistent labeling, and thus makes it difficult to learn a good predictor.

Chapter 6

Auto-Debiasing by Boosting a Biased Model

6.1 Introduction

In previous works, two heuristic assumptions are commonly used to train the bias-only model. The first is the "*weak-model*" assumption, which posits that models with lower capacity (e.g., Bag-of-words models or TinyBERT) are more likely to learn from the shallow heuristics of datasets and thus result in a bias-only model (Sanh et al., 2020). The second is the "*small-data*" assumption, which states that a model is prone to fitting shortcuts or biased features in the dataset during its early training stages (Utama et al., 2020b).

However, the assumptions used to train a bias-only model in previous works are uncertain and have many uncontrollable factors. It is difficult to define how weak the model should be or how small the dataset should be, resulting in redundant hyperparameters. Additionally, the bias-only model is inevitably fed with normal or robust samples due to both i) the unknown dataset-specific biasing sample proportion and ii) the randomness of model selection or data sampling. These uncontrollable factors can lead to a less-biased bias-only model, negatively impacting the learning of the debiased model.

Thereby, our goal is to develop a stable, automatic method for training a better biased model that is agnostic to the dataset, bias type, model size, and data scale. To achieve this, we conducted a pilot empirical study (see section §6.2), which aimed to identify the key factors for a better biased model. Our findings indicate that i) a higher proportion of bias in the training data results in a more biased model, and

ii) a more biased model has higher confidence in predicting bias.

This motivates us to propose a new debiasing framework, dubbed Bias-Progressive Auto-Debiasing (BiPAD), to obtain a better bias-only model by taking the inspiration of boosting learning. Specifically, our method alternates between biased data selection and bias-only model training, using the most biased samples from the previous step to train the bias-only model. Given our progressively-improved bias-boosted model that accurately identifies the biased samples, we can simply obtain a robust debiased model by a products-of-experts (PoE) loss (He et al., 2019).

We evaluate our approach in various settings and achieve significant improvement. To the best of our knowledge, our model delivers state-of-the-art performance on HANS (Zhang et al., 2019b), NLI Hard (Gururangan et al., 2018) and FEVER-SYMMETRIC (Schuster et al., 2019) without leveraging additional data. We plan to release the code as open-source after publication [*].

6.2 Empirical Study

Task Definition. We focus on natural language understanding (NLU) tasks and treat them as general multi-class classification problems. Given an input sentence pair $x \in X$, its goal is to predict the semantic relationship label $y \in \{1, 2, ..., K\}$, where K is the number of classes. Our goal is to obtain a robust debiased model F_d that can make predictions without relying on biased features $x_b \in x$ and instead focuses on unbiased features $x_u \in x$.

6.2.1 Insights about Debiasing Architectures

Debiasing architectures typically consist of two stages: first, a bias-only model F_b is constructed to calculate $P(y|x_b)$, which can be regarded as the confidence of a

[*] We will open our codes (uploaded), data, and models.

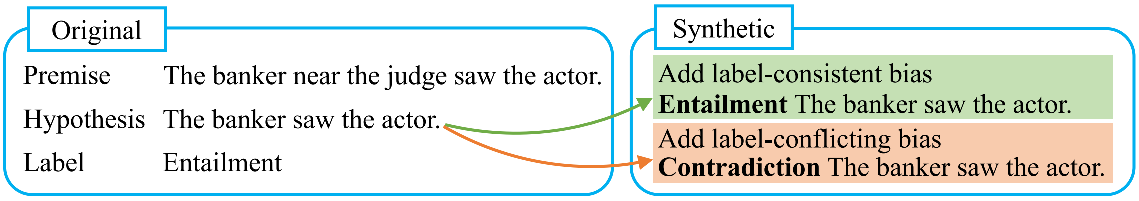

Original
Premise The banker near the judge saw the actor.
Hypothesis The banker saw the actor.
Label Entailment
Synthetic
Add label-consistent bias
Entailment The banker saw the actor.
Add label-conflicting bias
Contradiction The banker saw the actor.

Figure 6.1 : Example of the synthetic dataset, which is construct by inserting artificial shortcut in front of the hypothesis of original samples. Two types of synthetic bias, i.e., *label-consistent bias* and *label-conflicting bias*, are injected into the raw dataset.

sample being biased; then, a debiased model F_d is trained to reduce the importance of samples with high probability of being biased, thus behaving differently from the bias-only model.

Existing methods for building bias-only models are mainly based on the following observations: i) smaller models are more effective at learning bias information compared to larger models because biased features are more easily accessible than unbiased features (Sanh et al., 2020), and ii) a model can become biased if it is trained on a small fraction of the training dataset (Utama et al., 2020b). However, both observations do not guarantee a strongly biased model as they do not impose constraints on the dataset used to train the bias-only model, resulting in the model potentially learning general knowledge, especially on less-biased datasets.

In this work, we present a bias-progressive training strategy for obtaining a more biased bias-only model without the need for additional prior knowledge of dataset bias. The strategy is grounded on the following assumptions:

- *The more biased samples in training data, more biased the resulting model will be.*
- *Samples predicted by a bias-only model with high confidence are likely to be*

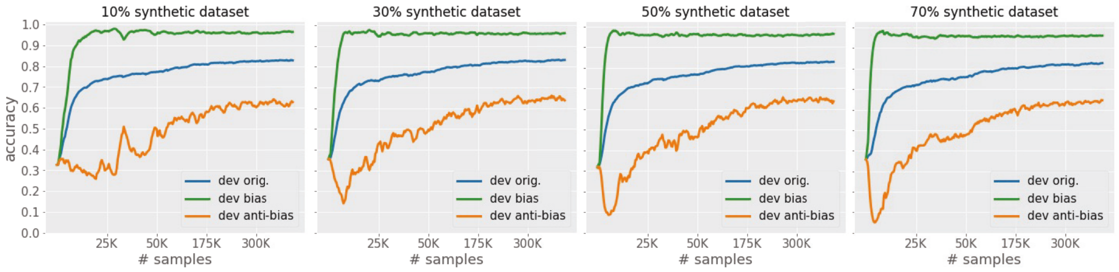


Figure 6.2 : Learning dynamics of BERT-base models fine-tuned on four synthetic MNLI training datasets with different $\eta \in \{0.1, 0.3, 0.5, 0.7\}$. All models are evaluated on three evaluation sets, the original MNLI dev set, the bias set, and the anti-bias set.

biased.

6.2.2 Verification by Synthetic Bias

Data Preparation. To validate these assumptions, we construct a controllable synthetic dataset by introducing artificial bias into the MNLI dataset (Williams et al., 2018) (additional information on the dataset can be found in Section 6.4.1). An example of the synthetic dataset can be seen in Figure 6.1. We simulated two types of bias by appending a specific string in front of the original hypothesis as a shortcut feature: One is *label-consistent bias*, which is constructed by inserting the golden label; The other is *label-conflicting bias*, where a random label other than the golden label is appended to the raw hypothesis sentence. Specifically, we added the synthesized bias to $\eta \in [0, 1]$ percentage of the training dataset. For each instance, the injected bias could either be a *label-consistent bias* or a *label-conflicting bias*, with a ratio of 8:2 to simulate the real-world distribution. Additionally, we also created two synthetic evaluation sets as a *label-consistent bias*-only set (*bias set*) and a *label-conflicting bias*-only set (*anti-bias set*). An ideal strong bias-only model should have learned the shortcuts, i.e., utilizing the inserted words as the

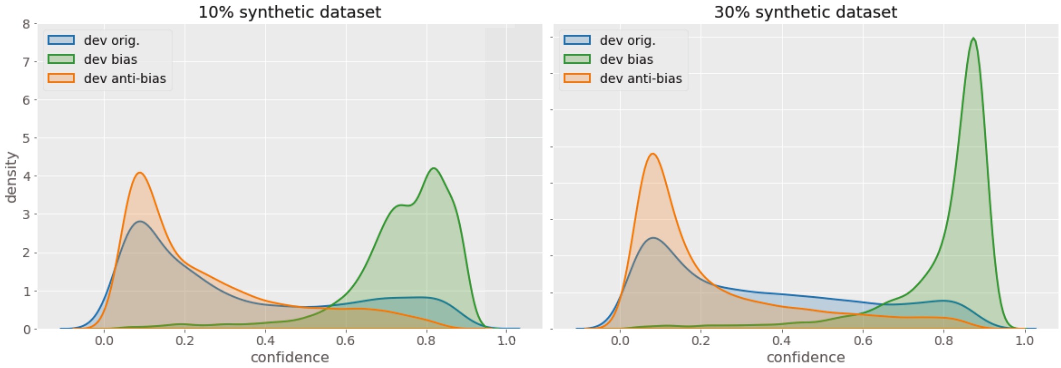


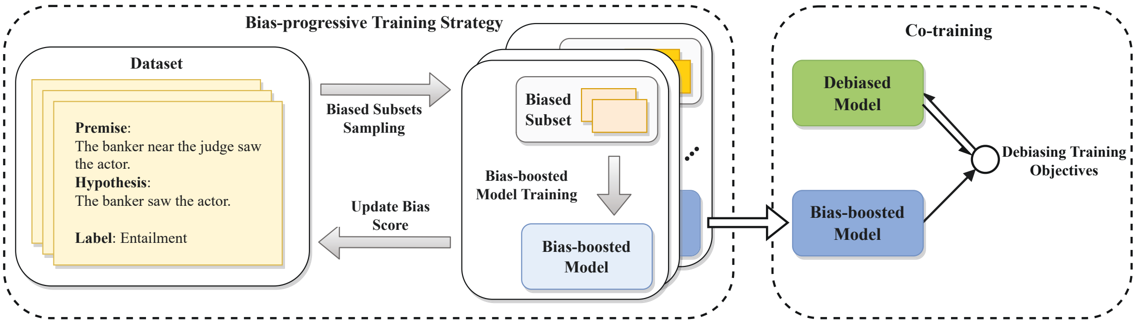
Figure 6.3 : The confidence distribution of samples on three evaluation sets. Models are trained with 2000 random samples in synthetic MNLI datasets with different $\eta \in \{0.1, 0.3\}$ for three epochs.

predictions. As such, it should have a significant performance gap when evaluated on the *bias set* and the *anti-bias set*.

Verifying our first assumption. We fine-tune a BERT-base model on several synthesized training datasets with different $\eta \in \{0.1, 0.3, 0.5, 0.7\}$ and evaluate them on three evaluation sets: the original MNLI evaluation set, the bias set, and the anti-bias set. As shown in Figure 6.2, at the early stages of the training process, the accuracy tends to increase to 100% on the *bias set* and drop to 0% on the *anti-bias set*, indicating that the language model is overfitting to superficial features in the first few training epochs, as also observed by Utama et al. (2020b). Furthermore, as the proportion of biased data η increases in the raw training data, the performance gap becomes more pronounced and stable between the bias set and the anti-bias set, resulting in a more biased model. This supports our first assumption that the more biased samples in the training data, the more biased a model will be.

Verifying our second assumption. We examine the bias-only model's confidence distribution on the three evaluation sets. As shown in Figure 6.3, the bias-only

Figure 6.4 : An overview of the bias-progressive auto-debiasing framework.



model makes predictions with high confidence on the *label-consistent bias* samples, but has low confidence in predicting the label for the *label-conflicting bias* samples. We can observe an apparent deviation in confidence among the three evaluation datasets even when the training dataset contains only a small fraction (i.e., 10%) of biased samples. This deviation becomes even more pronounced as the proportion of biased samples in the training dataset increases. This observation supports our assumptions that: i) the bias-only model will have high confidence in predicting the biased samples, and ii) such confidence increases as the model becomes more biased.

6.3 Bias-Progressive Auto-Debiasing

6.3.1 Overview

We propose a **Bi**as-**P**rogressive **A**uto-**D**ebiasing (BiPAD) framework for automatically and sufficiently training debiased models without the need for prior knowledge about biases. The framework, outlined in Figure 6.4, consists of: i) a bias-boosted model learned through a bias-progressive training strategy and ii) a robust debiased model co-trained with the fixed bias-boosted model on debiasing training objectives.

6.3.2 Bias-boosted Bias-only Model

Previous empirical studies reveal that a more biased bias-only model can be obtained by increasing the proportion of biased samples in the training dataset. To introduce more biased samples in the training data, we propose a bias-progressive training process that greedily learns from the most biased samples identified in the previous training step. Algorithm 1 outlines the steps to obtain a bias-boosted model. First, given a dataset $\mathbb{D}$ with N samples, we initialize bias scores $\{s_i|i \in (1, \ldots, N)\}$ for all samples $\{x_i|i \in (1, \ldots, N)\}$ as zero. At each step k, the weight w_i^k for x_i to be sampled is calculated by $w_i^k = \exp(s_i^k)/\sum_{j=1}^{N} \exp(s_j^k)$. We then sample a subset $D^k \subset \mathbb{D}$ with n instance based on the weights $\{w_i^k|i \in (1, \ldots, N)\}$ and train a bias-only model with the loss,

$$L_{\mathrm{CE}} = CrossEntropy(y, F_b^k(x, \theta_b^k)), \tag{6.1}$$

where θ_b^k stands for the parameters of the bias-only model. At the end of each step, we update the bias score for all samples with

$$s^{k+1} = \lambda s^k + (1 - \lambda) \cdot P(\hat{y}^t|x_i, \theta_b^k), \tag{6.2}$$

where $P(\hat{y}_i^t|x_i, \theta_b^k)$ is the confidence for model to predict true label of x_i, and λ is the moving average coefficient. In this way, samples detected as potentially biased (i,e, with high confidence) in this step will be more likely to be sampled as training data in the next step.

We repeat the above steps for K times to obtain the final bias-only model. At each step, we update the weights for the samples based on the confidence of the bias-only model. According to our *second observation*, the biased samples will have higher confidence scores and will be more likely to be selected as training data in the next step. As a result, the sampled subset in the next step will contain more biased samples, meaning that a more biased model F_b will be learned based on our *first observation.* In turn, a more biased model F_b will identify the biased samples more accurately. After the next update, the weights for biased samples will be more accurate and certain. By using this bias-progressive training strategy, we are able to obtain a strong bias-boosted model, even though the bias-only model is weak at the beginning.

6.3.3 Debiased Model Learning

Algorithm 1 Bias-progressive Training

```
Input: dataset $\mathbb{D}$ with $N$ samples; boost step $K$; subset size $n$; average coefficient $\lambda$
Output: bias-boosted model $F_b^K$
$s_i^1 \leftarrow 0 \ \forall i \in 1...N$
for $k \in 1...K$ do
    for $i \in 1...N$ do
        $w_i^k = \exp(s_i^k) / \sum_{j=1}^{N} \exp(s_j^k)$
    end for
    Sample $D^k \subset \mathbb{D}$ in size $n$ based on $w^k$
    Re-initialize pre-trained $F_b^k$
    Finetune $F_b^k$ on $D^k$ with cross-entropy
    for $i \in 1...N$ do
        $\Delta s_i = P(\hat{y}_i^t | x_i, \theta_b^k)$
        $s_i^{k+1} \leftarrow \lambda * s_i^k + (1 - \lambda) * \Delta s_i$
    end for
end for
```

After obtaining a bias-boosted bias-only model using the above steps, we then freeze the parameters of the bias-boosted model and train the debiased model through one of the two debiasing training objectives: example reweighting (Schuster et al., 2019) or product-of-experts (Sanh et al., 2020).

Example reweighting (ER) directly adjusts the weights of each training instance in the loss function based on the likelihood a training instance is biased, where the likelihood is obtained from the trained bias-boosted model F_b. The training objective for the debiased model F_d is:

$$L_{\text{ER}}{=}{-}\sum_{(x_i,y_i)\in\mathbb{D}}(1-P(\hat{y}_i^t|x_i,\theta_b))\log P(\hat{y}_i^t|x_i,\theta_d).$$

where $P(\hat{y}_i^t|x_i,\theta_b)$ is the confidence by the bias-only model for x_i to be its golden label, θ_d and θ_b are the parameters for the debiased model F_d and the bias-boosted model F_b, respectively.

Product-of-experts (PoE) encourages the debiased model to conpensate for the errors of the bias-boosted model, instead of sampling with frequently on the difficult samples. It learns the debiased model F_d via the following ensemble loss:

$$\begin{gathered}L_{\text{PoE}}=-\sum_{(x_i,y_i)\in\mathbb{D}}\log P(\hat{y}_i^t|x_i,\theta_d,\theta_b),\\ \text{where } P(\hat{y}_i|x_i,\theta_d,\theta_b)=\text{softmax}(\boldsymbol{l}_i^d+\boldsymbol{l}_i^b).\end{gathered}\tag{6.3}$$

Here, $\boldsymbol{l}^d$ and $\boldsymbol{l}^b$ indicate the logits obtained from the debiased model and the bias-boosted model, respectively.

In the following experiments, we primarily use the Product-of-experts (PoE) method to train the debiased model unless otherwise specified.

6.4 Experiments

6.4.1 Evaluation Datasets

We evaluate our proposed **Bi**as-**P**rogressive **A**uto-**D**ebiasing framework (BiPAD) on two real-world natural language understanding tasks, i.e., natural language inference and fact verification.

Natural language inference (NLI) tasks predict for the relationship between two sentences such as *entailment* and *contradiction*. We select the widely used Multi-Genre Natural Language Inference (MNLI) dataset (Williams et al., 2018) to train the bias-boosted model and the debiased model, then evaluate the performance of the debiased model on three evaluation datasets: MNLI-dev, HANS (Zhang et al., 2019b), and MNLI-Hard (Gururangan et al., 2018). MNLI dataset contains approximately 392K pairs of premises and hypotheses, labeled in three categories: *entailment*, *neutral* and *contradiction*. MNLI-dev is the original evaluation set for the MNLI dataset. HANS is a challenging test set for NLI tasks, which includes around 30K high word-overlapping sentence pairs generated by various templates, with each sample labeled as *entailment* or *non-entailment*, where the two types of labels are equally distributed. A known bias in the MNLI dataset is that high word-overlapping pairs are highly correlated with the label *entailment*, which makes a model easily perform poorly (predicting most samples as entailment) on HANS without debiasing strategies. MNLI-Hard (Gururangan et al., 2018) is a subset of MNLI-dev that consists of only challenging samples.

Fact verification tasks predict whether the evidence can support the given claim. The Fact Extraction and Verification (FEVER) dataset (Thorne et al., 2018) is commonly used for fact verification tasks, which consist of approximately 145K pairs of claims and evidence. Each pair is marked as *supporting*, *refuting*, or *insufficiently informative*. We use the FEVER dataset to train the bias-boosted model and the debiased model and evaluate the performance of the debiased model on two evaluation sets: FEVER-dev, the original evaluation set for FEVER, and SYMMETRIC (Schuster et al., 2019), a challenging test set synthesized based on the original sentence pairs in FEVER by inserting conflicting facts. Models that rely heavily on negation words such as "not" or "reject" will face a significant performance drop on this evaluation set.

6.4.2 Implementation Details

Based on the findings from our empirical study and the research by Utama et al. (2020), we train the bias-only model with a subset of 2000 samples for 3 epochs per iteration. In the bias-progressive training process, we set the number of iterations to 3, which has been verified to provide convergence. Additionally, we used a moving average coefficient of 0.5 as outlined in Section 6.3.2.

We fine-tuned both the bias-only model and the debiased model using the BERT-base model (Devlin et al., 2019) with a total of ∼110M parameters. The embedding size was set to 32, and the learning rate for both models was set to 2e-5. For the debiased model, the learning rate was linearly increased for 2000 warming steps and then linearly decreased to 0, whereas the learning rate for the bias-boosted model remained at 2e-5. We employed the Adam optimizer with its default hyperparameters. The implementation is based on Python and Hugging Face package, and trained on a RTX6000 GPU with Adam optimizer for approximately three hours.

6.4.3 Main Results

In this paragraph, we compare our proposed framework with a baseline BERT-base model trained with cross-entropy loss, as well as four existing state-of-the-art debiasing frameworks. Mahabadi et al. (2020) use prior knowledge of bias types to identify biased samples and train a bias-only model, then train a debiased model using product-of-experts (PoE). Utama et al. (2020b) obtain a bias-only model by training it on a small fraction of the dataset and train their debiased model through either PoE or example reweighting (ER). Sanh et al. (2020) train a BERT-tiny model on the entire dataset to obtain their bias-only model and use PoE to obtain the debiased model. Our proposed framework and the two latter works do not require prior knowledge about dataset bias. Table 6.1 compares the results of these methods on evaluation datasets. Results for comparison methods are taken from

Table 6.1 : Comparison results on the evaluation datasets in accuracy, where HANS-Ent, HANS-Non-Ent, and HANS-Total are the results for the *entailment* labeled samples, *non-entailment* labeled samples, and all samples, respectively.

	Objective	MNLI dev	HANS			MNLI Hard	FEVER	
			Total	Ent	Non-Ent		Dev	Symm.
BERT-base	CE	84.52	62.43	98.12	26.74	76.96	85.60	63.10
Mahabadi et al. (2020)	PoE	84.19	64.65	95.99	33.30	76.81	86.46	66.25
Utama et al. (2020b)	PoE	80.70	68.50	86.24	50.76	-	85.40	65.30
Utama et al. (2020b)	ER	81.40	68.60	87.06	50.14	-	87.20	65.60
Sanh et al. (2020)	PoE	81.35	68.77	81.13	56.41	76.54	-	-
BiPAD	ER	83.35	71.23	86.54	55.92	77.25	87.60	65.31
BiPAD	PoE	82.24	**73.82**	87.64	**60.20**	**77.48**	**87.80**	**66.62**

the original papers, and our results are an average of five trials.

The proposed BiPAD framework achieves state-of-the-art results on three challenging test sets, i.e., HANS, MNLI-Hard, and FEVER-Symm, outperforming previous results by 5.1%, 0.7%, and 0.4%, respectively. Specifically, it outperforms two prior knowledge-free frameworks on all three test sets by 5.1%, 1.0%, and 1.0%, indicating its superior performance in automatic bias capturing and debiasing. The framework also outperforms the framework utilizing manual prior knowledge, suggesting it captures unknown bias and exhibits strong generalization capability. Additionally, BiPAD consistently outperforms other works under different training objectives for the debiasing model, highlighting the effectiveness of its bias-progressive training strategy. Compared to the BERT-base model, all the debiasing frameworks show degradation on in-distribution datasets, i.e., MNLI-dev and HANS Ent, while BiPAD shows minimal reduction among all the frameworks. Overall, the proposed framework provides a stronger bias-boosted model and a robust debiased model.

6.4.4 Bias-boosted Model is More Biased

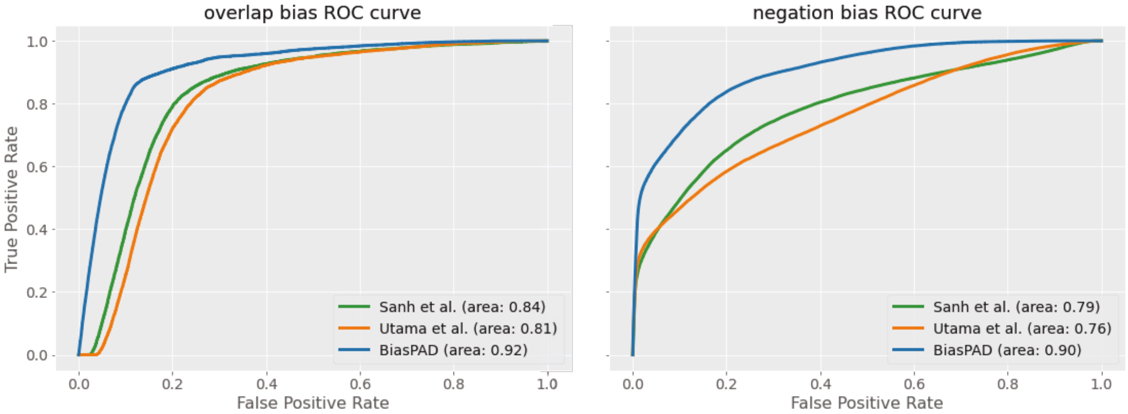


Figure 6.5 : AUC-ROC curve on datasets synthesized for two known biases on MNLI-dev dataset by three bias-only models trained with different strategies.

In Section 6.4.3, we present evidence that our proposed bias-progressive training strategy produces a stronger bias-only model compared to the methods proposed by Utama et al. (2020b) and Sanh et al. (2020). To visualize this difference, we conduct experiments in which we obtain three bias-only models using the strategies proposed by Utama et al. (2020b), Sanh et al. (2020), and our bias-progressive training strategy. We reproduce the other two methods using the suggested hyper-parameters from their respective papers and provide the details in the Appendix. We then synthesize two evaluation sets based on the MNLI-dev set. The first one relabels samples as 1 if it has a high word overlap rate, and the original label is *entailment* and as 0 for all other samples. The second one relabels samples as 1 if it contains negation words in the hypothesis, and the original label is *contradiction* and as 0 for all other samples. The calculations for the word overlap rate and the list of negation words are also in the Appendix. We evaluate the performance of the three bias-only models on these two synthesized datasets and present the results in Figure 6.5 in the form of AUC-ROC curves based on the confidence of the bias-only models. The results show that our bias-progressive training strategy outperforms the other two on both types of biases, with higher AUC scores and dominant ROC curves. This experiment demonstrates that our bias-boosted model has a stronger ability to discriminate between the two well-known biases compared to the others and is, therefore a more biased model.

6.4.5 Number of Iterations to Obtain the Best Bias-Boosted Model

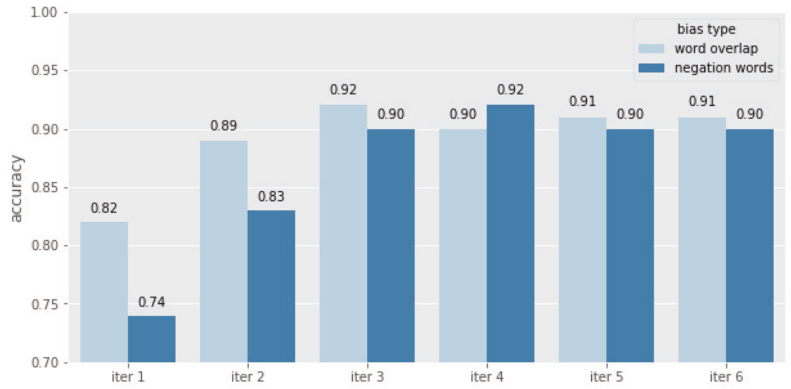


Figure 6.6 : AUC scores for two datasets synthesized for two known biases on MNLI-dev dataset by our bias-boosted models at different iterations.

We developed a bias-boosted model using a bias-progressive training strategy, training the bias-only model step-by-step. One question we aimed to answer was how many iterations were necessary to achieve the best bias-boosted model. To answer this, we conducted an experiment to observe the model's convergence during the bias-progressive training process. We verified the results on two synthesized biased datasets same as the datasets in section 6.4.4. We iterated the bias-progressive training for six steps, evaluating the bias-only model on the two datasets and recording their AUC scores after each step. The results, shown in Figure 6.6, show that the AUC scores increase from 0.82/0.74 to 0.92/0.90 for the two evaluation sets in the first three iterations and only show slight fluctuations after the fourth iteration. We conclude that the bias-boosted model converges through the bias-progressive training process in the first few iterations. Therefore, we selected $K = 3$ iterations to obtain the bias-boosted model in our experiments.

6.4.6 Trade-off between In-distribution and Out-of-distribution Sets

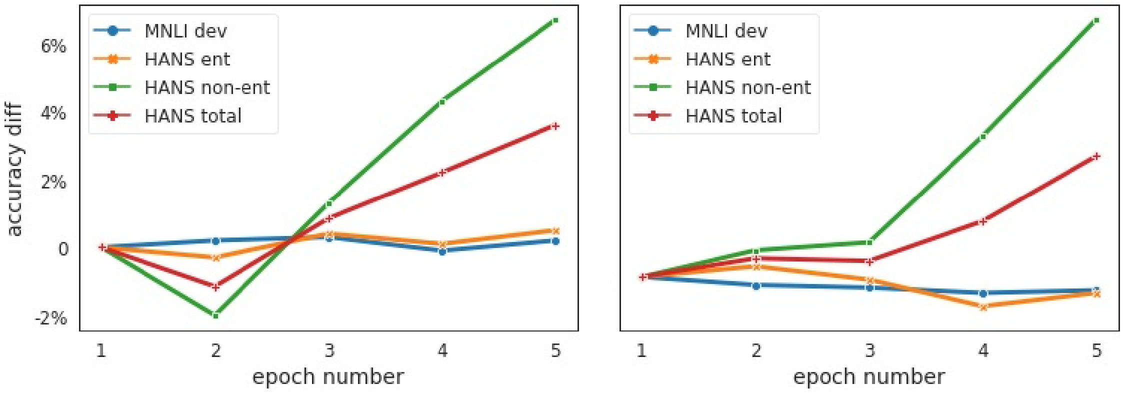


Figure 6.7 : The x-axis indicates for the t-th epoch. **Left**: Accuracy difference of injecting cross-entropy loss at t-th epoch for only one epoch. **Right**: Accuracy difference of injecting cross-entropy loss starting at t-th epoch. For a clearer contrast, we show the difference value against the leftmost point.

In this study, we investigate the trade-off between in-domain and out-of-domain performance of a debiased model by setting the training objective as the following multi-loss function

$$L = L_{\mathrm{PoE}} + \alpha L_{\mathrm{CE}}, \tag{6.4}$$

where L_{CE} is a normal cross-entropy loss and α is the parameter used to adjust the trade-off. Intuitively, if a BERT-base model is fine-tuned with only the cross-entropy loss, it will result in a biased BERT-base model that is similar to the baseline BERT-base model compared in Section 6.1. One advantage of introducing the cross-entropy loss is to improve in-distribution performance, as we have observed a performance drop on in-distribution datasets when using debiasing strategies. Therefore, our question is whether it is possible to obtain a debiased model that performs well in both in- and out-of-distribution scenarios by training it with the objective in Equation 6.4. We answer this question by observing the performance trade-off in two strategies: 1) inserting the cross-entropy loss at the t-th training epoch, and 2) continually inserting the cross-entropy loss from the t-th training epoch.

Figure 6.7 illustrates the performance of the two strategies on four evaluation datasets. MNLI-dev and HANS-ent are considered as in-distribution sets, while HANS-not-ent is an out-of-distribution set, and HANS-total is the overall performance. Both strategies demonstrate that adding the CE loss at a later stage improves in-distribution performance while preserving out-of-distribution performance better. Although using the CE loss will still harm out-of-distribution performance, a better trade-off between in- and out-of-distribution performance can be achieved by adding the CE loss in a later stage of the debiased model training process.

Chapter 7

Conclusion and Future Directions

In summarizing the contributions of this thesis, we embarked on an in-depth exploration of augmenting machine reading comprehension models, delving into four pivotal areas:

- Augmenting knowledge graphs via an innovative graph embedding technique.
- Enhancing procedural text understanding through the development of a unique entity-action-location graph model.
- Advancing temporal reading comprehension by applying a detailed question understanding strategy.
- Implementing strategies to mitigate dataset bias in natural language understanding tasks.

Our novel Relation-adaptive Translating Embedding (RatE) for knowledge graph completion emerged as a significant advancement, simultaneously improving representation and modeling capacities while addressing the embedding ambiguity problem presented by non-injective relations. RatE distinguishes itself by striking an optimal balance between interpretability and expressive prowess, surpassing preceding trans-based methodologies. Additionally, it cohesively integrates with a local-cognitive negative sampling method, showcasing state-of-the-art performance on four link prediction benchmarks.

The Real methodology, tailored for procedural text understanding, was yet another milestone. We pioneered an entity-action-location graph to holistically model diverse concepts and interrelations. Validated against formidable baselines on two benchmark datasets, the efficacy of our approach is clear. Looking ahead, we aim to incorporate entity disambiguation and external knowledge to elevate procedural text comprehension. Considering the prowess of Large Language Models (LLMs) in comprehending and generating human-like text, pairing them with Knowledge Graphs (KGs) offers potential advancements in question answering. If a KG lacks specific information, an LLM can predict or infer the missing links, enhancing the robustness and accuracy of the Q&A system.

For the challenges inherent in temporal reading comprehension, we devised an innovative method that adeptly encodes temporal ordering queries into pertinent events and related temporal dynamics. To discern the nuances among temporal relations, this method also utilizes a contrastive loss mechanism. Our future endeavors will address a broader range of temporal relation understanding issues and bolster the overall comprehension of passages related to temporal reading. Addressing the intricate realm of temporal-spatial relations and their dynamics in process comprehension, we advocate for the utility of graphs as natural mediums to portray relationships. Transforming text into graphs empowers LLMs with superior relational reasoning abilities, fostering an enhanced understanding of the interplay among various entities.

Our cutting-edge BiPAD framework was formulated to curb dataset bias in natural language understanding tasks. Rooted in a bias-progressive training paradigm, it effectively delivers both a bias-amplified and a debiased model. BiPAD's exemplary performance across three rigorous datasets stands as testimony to its prowess. As we proceed, we're keen on exploring synergies between the bias-only and debiased models to further refine performance. Contrastive learning's unique approach, which differentiates between similar temporal sequences, augments an LLM's discernment of subtle variations in sequences. This amplifies the model's grasp of temporal data nuances, with potential benefits across diverse tasks. A debiased LLM promises users and organizations greater trustworthiness by ensuring outputs devoid of harmful stereotypes or biases. Removing these biases paves the way for models to generalize more efficiently across tasks, sidestepping overfitting to any skewed nuances present in the training dataset.

Another future direction is combining Federated Learning (FL) (Tan et al., 2022, 2023a,b; Ma et al., 2023; Gupta et al., 2022; Chen et al., 2023) with Natural Language Processing (NLP). Federated Learning presents a transformative approach for enhancing Language Models (LMs), such as those utilized in NLP. By enabling distributed model training across multiple devices while keeping the data localized, FL addresses significant concerns related to privacy and data security. This methodology allows LMs to learn from diverse datasets without the need to centralize sensitive information, thereby enriching the model's understanding and performance across various languages and dialects. In the context of LMs, applying Federated Learning can significantly improve the model's ability to understand context, semantics, and subtleties in language by leveraging data from a wide range of sources, each contributing unique linguistic patterns and usage scenarios. As a result, Federated Learning not only fortifies data privacy but also enhances the robustness and inclusivity of language models, making them more reflective of the global diversity in language usage.

To encapsulate, this thesis delineates landmark methods in the realm of Natural Language Understanding, setting the stage for richer comprehension and reasoning capacities in machine reading models. We view knowledge graphs and debiasing models as overarching strategies to boost LLMs performance. In contrast, procedural text understanding and temporal relation extraction serve as specialized tactics to optimize LLMs performance for distinct datasets. These advancements would result in LLMs that are not only more knowledgeable and understanding of complex textual content but also fairer and more aligned with ethical standards, thus broadening their applicability and impact in real-world scenarios. This body of work lays a robust groundwork for upcoming explorations in the field.

Bibliography

Amini, A., Bosselut, A., Mishra, B. D., Choi, Y. & Hajishirzi, H., 2020, 'Procedural reading comprehension with attribute-aware context flow', Das, D., Hajishirzi, H., McCallum, A. & Singh, S. (eds.) *Conference on Automated Knowledge Base Construction, AKBC 2020, Virtual, June 22-24, 2020*, <`https://doi.org/10.24432/C5C883`>.

Balazevic, I., Allen, C. & Hospedales, T. M., 2019, 'Tucker: Tensor factorization for knowledge graph completion', Inui, K., Jiang, J., Ng, V. & Wan, X. (eds.) *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, Association for Computational Linguistics, pp. 5184–5193, <`https://doi.org/10.18653/v1/D19-1522`>.

Bollacker, K. D., Evans, C., Paritosh, P., Sturge, T. & Taylor, J., 2008, 'Freebase: a collaboratively created graph database for structuring human knowledge', Wang, J. T. (ed.) *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*, ACM, pp. 1247–1250, <`https://doi.org/10.1145/1376616.1376746`>.

Bordes, A., Usunier, N., García-Durán, A., Weston, J. & Yakhnenko, O., 2013, 'Translating embeddings for modeling multi-relational data', Burges, C. J. C., Bottou, L., Ghahramani, Z. & Weinberger, K. Q. (eds.) *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake*

Tahoe, Nevada, United States, pp. 2787–2795, <http://papers.nips.cc/paper/5071-translating-embeddings-for-modeling-multi-relational-data>.

Bosselut, A., Levy, O., Holtzman, A., Ennis, C., Fox, D. & Choi, Y., 2018, 'Simulating action dynamics with neural process networks', *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, OpenReview.net, <https://openreview.net/forum?id=rJYFzMZC->.

Cai, L. & Wang, W. Y., 2018, 'KBGAN: adversarial learning for knowledge graph embeddings', Walker, M. A., Ji, H. & Stent, A. (eds.) *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, Association for Computational Linguistics, pp. 1470–1480, <https://doi.org/10.18653/v1/n18-1133>.

Cassidy, T., McDowell, B., Chambers, N. & Bethard, S., 2014, 'An annotation framework for dense event ordering', *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 2: Short Papers*, The Association for Computer Linguistics, pp. 501–506.

Chen, C., Feng, X., Zhou, J., Yin, J. & Zheng, X., 2023, 'Federated large language model: A position paper', *CoRR*, vol. abs/2307.08925, <https://doi.org/10.48550/arXiv.2307.08925>.

Chen, K., Xu, W., Cheng, X., Xiaochuan, Z., Zhang, Y., Song, L., Wang, T., Qi, Y. & Chu, W., 2020, 'Question directed graph attention network for numerical reasoning over text', Webber, B., Cohn, T., He, Y. & Liu, Y. (eds.) *Proceedings* *of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, Association for Computational Linguistics, pp. 6759–6768.

Cheng, F., Asahara, M., Kobayashi, I. & Kurohashi, S., 2020, 'Dynamically updating event representations for temporal relation classification with multi-category learning', Cohn, T., He, Y. & Liu, Y. (eds.) *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, , vol. EMNLP 2020 of *Findings of ACL*Association for Computational Linguistics, pp. 1352–1357.

Clark, C., Yatskar, M. & Zettlemoyer, L., 2019, 'Don't take the easy way out: Ensemble based methods for avoiding known dataset biases', Inui, K., Jiang, J., Ng, V. & Wan, X. (eds.) *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, Association for Computational Linguistics, pp. 4067–4080, <https://doi.org/10.18653/v1/D19-1418>.

Dalvi, B., Huang, L., Tandon, N., Yih, W. & Clark, P., 2018, 'Tracking state changes in procedural text: a challenge dataset and models for process paragraph comprehension', Walker, M. A., Ji, H. & Stent, A. (eds.) *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, Association for Computational Linguistics, pp. 1595–1604, <https://doi.org/10.18653/v1/n18-1144>.

Dalvi, B., Tandon, N., Bosselut, A., Yih, W. & Clark, P., 2019, 'Everything happens for a reason: Discovering the purpose of actions in procedural text', Inui, K., Jiang, J., Ng, V. & Wan, X. (eds.) *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, Association for Computational Linguistics, pp. 4495–4504, <https://doi.org/10.18653/v1/D19-1457>.

Das, R., Munkhdalai, T., Yuan, X., Trischler, A. & McCallum, A., 2019, 'Building dynamic knowledge graphs from text using machine reading comprehension', *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, OpenReview.net, <https://openreview.net/forum?id=S1lhbnRqF7>.

Dettmers, T., Minervini, P., Stenetorp, P. & Riedel, S., 2017, 'Convolutional 2d knowledge graph embeddings', *CoRR*, vol. abs/1707.01476, <http://arxiv.org/abs/1707.01476>.

Devlin, J., Chang, M., Lee, K. & Toutanova, K., 2019, 'BERT: pre-training of deep bidirectional transformers for language understanding', Burstein, J., Doran, C. & Solorio, T. (eds.) *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, Association for Computational Linguistics, pp. 4171–4186.

Dhingra, B., Liu, H., Yang, Z., Cohen, W. W. & Salakhutdinov, R., 2017, 'Gated-attention readers for text comprehension', Barzilay, R. & Kan, M. (eds.) *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, Association for Computational Linguistics, pp. 1832–1846.

Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., Gao, J., Zhou, M. & Hon, H., 2019, 'Unified language model pre-training for natural language understanding and generation', Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B. & Garnett, R. (eds.) *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 13042–13054.

Du, X., Mishra, B. D., Tandon, N., Bosselut, A., Yih, W., Clark, P. & Cardie, C., 2019, 'Be consistent! improving procedural text comprehension using label consistency', Burstein, J., Doran, C. & Solorio, T. (eds.) *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, Association for Computational Linguistics, pp. 2347–2356, <https://doi.org/10.18653/v1/n19-1244>.

Dua, D., Wang, Y., Dasigi, P., Stanovsky, G., Singh, S. & Gardner, M., 2019, 'DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs', Burstein, J., Doran, C. & Solorio, T. (eds.) *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, Association for Computational Linguistics, pp. 2368–2378.

Durrett, G. & Klein, D., 2015, 'Neural CRF parsing', *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China,* *Volume 1: Long Papers*, The Association for Computer Linguistics, pp. 302–312, <https://doi.org/10.3115/v1/p15-1030>.

Ebisu, T. & Ichise, R., 2018, 'Toruse: Knowledge graph embedding on a lie group', McIlraith, S. A. & Weinberger, K. Q. (eds.) *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, AAAI Press, pp. 1819–1826, <https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16227>.

Gao, T., Yao, X. & Chen, D., 2021, 'Simcse: Simple contrastive learning of sentence embeddings', *CoRR*, vol. abs/2104.08821.

Gupta, A. & Durrett, G., 2019, 'Tracking discrete and continuous entity state for process understanding', Martins, A. F. T., Vlachos, A., Kozareva, Z., Ravi, S., Lampouras, G., Niculae, V. & Kreutzer, J. (eds.) *Proceedings of the Third Workshop on Structured Prediction for NLP@NAACL-HLT 2019, Minneapolis, Minnesota, Jun 7, 2019*, Association for Computational Linguistics, pp. 7–12, <https://doi.org/10.18653/v1/W19-1502>.

Gupta, S., Huang, Y., Zhong, Z., Gao, T., Li, K. & Chen, D., 2022, 'Recovering private text in federated learning of language models', Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K. & Oh, A. (eds.) *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, <http://papers.nips.cc/paper_files/paper/2022/hash/35b5c175e139bff5f22a5361270fce87-Abstract-Conference.html>.

Gururangan, S., Swayamdipta, S., Levy, O., Schwartz, R., Bowman, S. R. & Smith, N. A., 2018, 'Annotation artifacts in natural language inference data', Walker,

M. A., Ji, H. & Stent, A. (eds.) *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, Association for Computational Linguistics, pp. 107–112, <`https://doi.org/10.18653/v1/n18-2017`>.

Hao, Y., Zhang, Y., Liu, K., He, S., Liu, Z., Wu, H. & Zhao, J., 2017, 'An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge', Barzilay, R. & Kan, M. (eds.) *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, Association for Computational Linguistics, pp. 221–231, <`https://doi.org/10.18653/v1/P17-1021`>.

He, H., Zha, S. & Wang, H., 2019, 'Unlearn dataset bias in natural language inference by fitting the residual', *EMNLP-IJCNLP 2019*, p. 132.

Henaff, M., Weston, J., Szlam, A., Bordes, A. & LeCun, Y., 2017, 'Tracking the world state with recurrent entity networks', *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, OpenReview.net, <`https://openreview.net/forum?id=rJTKKKqeg`>.

Hochreiter, S. & Schmidhuber, J., 1997, 'Long short-term memory', *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, <`https://doi.org/10.1162/neco.1997.9.8.1735`>.

Huang, H., Geng, X., Long, G. & Jiang, D., 2022, 'Understand before answer: Improve temporal reading comprehension via precise question understanding', Carpuat, M., de Marneffe, M. & Ruíz, I. V. M. (eds.) *Proceedings of the 2022*

Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022, Association for Computational Linguistics, pp. 375–384, <https://doi.org/10.18653/v1/2022.naacl-main.28>.

Huang, H., Geng, X., Pei, J., Long, G. & Jiang, D., 2021, 'Reasoning over entity-action-location graph for procedural text understanding', Zong, C., Xia, F., Li, W. & Navigli, R. (eds.) *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, Association for Computational Linguistics, pp. 5100–5109, <https://doi.org/10.18653/v1/2021.acl-long.396>.

Huang, H., Long, G., Shen, T., Jiang, J. & Zhang, C., 2020, 'Rate: Relation-adaptive translating embedding for knowledge graph completion', Scott, D., Bel, N. & Zong, C. (eds.) *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, International Committee on Computational Linguistics, pp. 556–567, <https://doi.org/10.18653/v1/2020.coling-main.48>.

Ji, G., He, S., Xu, L., Liu, K. & Zhao, J., 2015, 'Knowledge graph embedding via dynamic mapping matrix', *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, The Association for Computer Linguistics, pp. 687–696, <https://doi.org/10.3115/v1/p15-1067>.

Kadlec, R., Schmid, M., Bajgar, O. & Kleindienst, J., 2016, 'Text understanding with the attention sum reader network', *Proceedings of the 54th Annual Meet-*

ing of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers, The Association for Computer Linguistics.

Krompaß, D., Baier, S. & Tresp, V., 2015, 'Type-constrained representation learning in knowledge graphs', *CoRR*, vol. abs/1508.02593, <`http://arxiv.org/abs/1508.02593`>.

Le Bras, R., Swayamdipta, S., Bhagavatula, C., Zellers, R., Peters, M. E., Sabharwal, A. & Choi, Y., 2020, 'Adversarial filters of dataset biases', *ICML*, .

Li, Y., Long, G., Shen, T., Zhou, T., Yao, L., Huo, H. & Jiang, J., 2020, 'Self-attention enhanced selective gate with entity-aware embedding for distantly supervised relation extraction', *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, AAAI Press, pp. 8269–8276, <`https://doi.org/10.1609/aaai.v34i05.6342`>.

Lin, Y., Liu, Z., Sun, M., Liu, Y. & Zhu, X., 2015, 'Learning entity and relation embeddings for knowledge graph completion', Bonet, B. & Koenig, S. (eds.) *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, AAAI Press, pp. 2181–2187, <`http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9571`>.

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', *CoRR*, vol. abs/1907.11692.

Ma, J., Zhou, T., Long, G., Jiang, J. & Zhang, C., 2023, 'Structured federated learning through clustered additive modeling', Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M. & Levine, S. (eds.) *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, <http://papers.nips.cc/paper_files/paper/2023/hash/8668fdc7b2ddf55a0e235824c66f2eee-Abstract-Conference.html>.

Mahabadi, R. K., Belinkov, Y. & Henderson, J., 2020, 'End-to-end bias mitigation by modelling biases in corpora', Jurafsky, D., Chai, J., Schluter, N. & Tetreault, J. R. (eds.) *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, Association for Computational Linguistics, pp. 8706–8716, <https://doi.org/10.18653/v1/2020.acl-main.769>.

McCoy, T., Pavlick, E. & Linzen, T., 2019, 'Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference', Korhonen, A., Traum, D. R. & Màrquez, L. (eds.) *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, Association for Computational Linguistics, pp. 3428–3448, <https://doi.org/10.18653/v1/p19-1334>.

Miller, G. A., 1995, 'Wordnet: A lexical database for english', *Commun. ACM*, vol. 38, no. 11, pp. 39–41, <http://doi.acm.org/10.1145/219717.219748>.

Min, B., Grishman, R., Wan, L., Wang, C. & Gondek, D., 2013, 'Distant supervision for relation extraction with an incomplete knowledge base', Vanderwende, L., III, H. D. & Kirchhoff, K. (eds.) *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia,* *USA*, The Association for Computational Linguistics, pp. 777–782, <https://www.aclweb.org/anthology/N13-1095/>.

Nickel, M., Rosasco, L. & Poggio, T. A., 2016, ‘Holographic embeddings of knowledge graphs’, Schuurmans, D. & Wellman, M. P. (eds.) *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, AAAI Press, pp. 1955–1961, <http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12484>.

Nickel, M., Tresp, V. & Kriegel, H., 2011, ‘A three-way model for collective learning on multi-relational data’, Getoor, L. & Scheffer, T. (eds.) *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, Omnipress, pp. 809–816, <https://icml.cc/2011/papers/438_icmlpaper.pdf>.

Nie, Y., Williams, A., Dinan, E., Bansal, M., Weston, J. & Kiela, D., 2020, ‘Adversarial NLI: A new benchmark for natural language understanding’, Jurafsky, D., Chai, J., Schluter, N. & Tetreault, J. R. (eds.) *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, Association for Computational Linguistics, pp. 4885–4901, <https://doi.org/10.18653/v1/2020.acl-main.441>.

Ning, Q., Feng, Z., Wu, H. & Roth, D., 2019, ‘Joint reasoning for temporal and causal relations’, *CoRR*, vol. abs/1906.04941.

Ning, Q., Wu, H., Han, R., Peng, N., Gardner, M. & Roth, D., 2020, ‘Torque: A reading comprehension dataset of temporal ordering questions’, *arXiv preprint arXiv:2005.00242*.

Ning, Q., Wu, H. & Roth, D., 2018, ‘A multi-axis annotation scheme for event temporal relations’, Gurevych, I. & Miyao, Y. (eds.) *Proceedings of the 56th Annual*

Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers, Association for Computational Linguistics, pp. 1318–1328.

Poliak, A., Naradowsky, J., Haldar, A., Rudinger, R. & Durme, B. V., 2018, 'Hypothesis only baselines in natural language inference', Nissim, M., Berant, J. & Lenci, A. (eds.) *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics, *SEM@NAACL-HLT 2018, New Orleans, Louisiana, USA, June 5-6, 2018*, Association for Computational Linguistics, pp. 180–191, <`https://doi.org/10.18653/v1/s18-2023`>.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W. & Liu, P. J., 2020, 'Exploring the limits of transfer learning with a unified text-to-text transformer', *J. Mach. Learn. Res.*, vol. 21, pp. 140:1–140:67.

Rajpurkar, P., Jia, R. & Liang, P., 2018, 'Know what you don't know: Unanswerable questions for squad', Gurevych, I. & Miyao, Y. (eds.) *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, Association for Computational Linguistics, pp. 784–789.

Rajpurkar, P., Zhang, J., Lopyrev, K. & Liang, P., 2016, 'Squad: 100, 000+ questions for machine comprehension of text', Su, J., Carreras, X. & Duh, K. (eds.) *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, The Association for Computational Linguistics, pp. 2383–2392.

Sanh, V., Wolf, T., Belinkov, Y. & Rush, A. M., 2020, 'Learning from others' mistakes: Avoiding dataset biases without modeling them', *International Conference on Learning Representations*, .

Saxon, M., Wang, X., Xu, W. & Wang, W. Y., 2022, 'Peco: Examining single sentence label leakage in natural language inference datasets through progressive evaluation of cluster outliers', <`https://arxiv.org/abs/2112.09237`>.

Schlichtkrull, M. S., Kipf, T. N., Bloem, P., van den Berg, R., Titov, I. & Welling, M., 2018, 'Modeling relational data with graph convolutional networks', Gangemi, A., Navigli, R., Vidal, M., Hitzler, P., Troncy, R., Hollink, L., Tordai, A. & Alam, M. (eds.) *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, , vol. 10843 of *Lecture Notes in Computer Science*Springer, pp. 593–607, <`https://doi.org/10.1007/978-3-319-93417-4\_38`>.

Schuster, T., Shah, D., Yeo, Y. J. S., Ortiz, D. R. F., Santus, E. & Barzilay, R., 2019, 'Towards debiasing fact verification models', *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3419–3425.

Seo, M. J., Min, S., Farhadi, A. & Hajishirzi, H., 2017, 'Query-reduction networks for question answering', *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, OpenReview.net, <`https://openreview.net/forum?id=B1MRcPclx`>.

Shen, T., Zhou, T., Long, G., Jiang, J., Pan, S. & Zhang, C., 2018a, 'Disan: Directional self-attention network for rnn/cnn-free language understanding', McIlraith, S. A. & Weinberger, K. Q. (eds.) *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA,* *February 2-7, 2018*, AAAI Press, pp. 5446–5455, <https://doi.org/10.1609/aaai.v32i1.11941>.

Shen, T., Zhou, T., Long, G., Jiang, J. & Zhang, C., 2018b, 'Bi-directional block self-attention for fast and memory-efficient sequence modeling', *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, OpenReview.net, <https://openreview.net/forum?id=H1cWzoxA->.

Shi, P. & Lin, J., 2019, 'Simple BERT models for relation extraction and semantic role labeling', *CoRR*, vol. abs/1904.05255, <http://arxiv.org/abs/1904.05255>.

Speer, R., Chin, J. & Havasi, C., 2017, 'Conceptnet 5.5: An open multilingual graph of general knowledge', Singh, S. P. & Markovitch, S. (eds.) *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, AAAI Press, pp. 4444–4451.

Sun, Z., Deng, Z., Nie, J. & Tang, J., 2019, 'Rotate: Knowledge graph embedding by relational rotation in complex space', *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, <https://openreview.net/forum?id=HkgEQnRqYQ>.

Tan, Y., Chen, C., Zhuang, W., Dong, X., Lyu, L. & Long, G., 2023a, 'Is heterogeneity notorious? taming heterogeneity to handle test-time shift in federated learning', Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M. & Levine, S. (eds.) *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, <http://papers.nips.cc/paper_files/paper/2023/hash/565f995643da6329cec701f26f8579f5-Abstract-Conference.html>.

Tan, Y., Liu, Y., Long, G., Jiang, J., Lu, Q. & Zhang, C., 2023b, 'Federated learning on non-iid graphs via structural knowledge sharing', Williams, B., Chen, Y. & Neville, J. (eds.) *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, AAAI Press, pp. 9953–9961, <https://doi.org/10.1609/aaai.v37i8.26187>.

Tan, Y., Long, G., Ma, J., Liu, L., Zhou, T. & Jiang, J., 2022, 'Federated learning from pre-trained models: A contrastive learning approach', Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K. & Oh, A. (eds.) *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, <http://papers.nips.cc/paper_files/paper/2022/hash/7aa320d2b4b8f6400b18f6f77b6c1535-Abstract-Conference.html>.

Tandon, N., Dalvi, B., Grus, J., Yih, W., Bosselut, A. & Clark, P., 2018, 'Reasoning about actions and state changes by injecting commonsense knowledge', Riloff, E., Chiang, D., Hockenmaier, J. & Tsujii, J. (eds.) *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, Association for Computational Linguistics, pp. 57–66, <https://doi.org/10.18653/v1/d18-1006>.

Thorne, J., Vlachos, A., Christodoulopoulos, C. & Mittal, A., 2018, 'FEVER: a large-scale dataset for fact extraction and verification', Walker, M. A., Ji, H. & Stent, A. (eds.) *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1*

(Long Papers), Association for Computational Linguistics, pp. 809–819, <https://doi.org/10.18653/v1/n18-1074>.

Toutanova, K., Chen, D., Pantel, P., Poon, H., Choudhury, P. & Gamon, M., 2015, ‘Representing text for joint embedding of text and knowledge bases’, Màrquez, L., Callison-Burch, C., Su, J., Pighin, D. & Marton, Y. (eds.) *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, The Association for Computational Linguistics, pp. 1499–1509, <https://doi.org/10.18653/v1/d15-1174>.

Trouillon, T., Welbl, J., Riedel, S., Gaussier, É. & Bouchard, G., 2016, ‘Complex embeddings for simple link prediction’, Balcan, M. & Weinberger, K. Q. (eds.) *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, , vol. 48 of *JMLR Workshop and Conference Proceedings*JMLR.org, pp. 2071–2080, <http://proceedings.mlr.press/v48/trouillon16.html>.

Tsuchiya, M., 2018, ‘Performance impact caused by hidden bias of training data for recognizing textual entailment’, Calzolari, N., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Hasida, K., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S. & Tokunaga, T. (eds.) *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*, European Language Resources Association (ELRA), <http://www.lrec-conf.org/proceedings/lrec2018/summaries/786.html>.

Utama, P. A., Moosavi, N. S. & Gurevych, I., 2020a, ‘Mind the trade-off: Debiasing NLU models without degrading the in-distribution performance’, Jurafsky, D., Chai, J., Schluter, N. & Tetreault, J. R. (eds.) *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July*

5-10, 2020, Association for Computational Linguistics, pp. 8717–8729, <https://doi.org/10.18653/v1/2020.acl-main.770>.

Utama, P. A., Moosavi, N. S. & Gurevych, I., 2020b, ‘Towards debiasing NLU models from unknown biases’, Webber, B., Cohn, T., He, Y. & Liu, Y. (eds.) *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, Association for Computational Linguistics, pp. 7597–7610, <https://doi.org/10.18653/v1/2020.emnlp-main.613>.

UzZaman, N., Llorens, H., Derczynski, L., Allen, J. F., Verhagen, M. & Pustejovsky, J., 2013, ‘Semeval-2013 task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations’, Diab, M. T., Baldwin, T. & Baroni, M. (eds.) *Proceedings of the 7th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2013, Atlanta, Georgia, USA, June 14-15, 2013*, The Association for Computer Linguistics, pp. 1–9.

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P. & Bengio, Y., 2018, ‘Graph attention networks’, *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, OpenReview.net, <https://openreview.net/forum?id=rJXMpikCZ>.

Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O. & Bowman, S. R., 2019a, ‘Superglue: A stickier benchmark for general-purpose language understanding systems’, Wallach, H. M., Larochelle, H., Beygelzimer, A., d’Alché-Buc, F., Fox, E. B. & Garnett, R. (eds.) *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver,* *BC, Canada*, pp. 3261–3275, <https://proceedings.neurips.cc/paper/2019/hash/4496bf24afe7fab6f046bf4923da8de6-Abstract.html>.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O. & Bowman, S. R., 2018, 'GLUE: A multi-task benchmark and analysis platform for natural language understanding', Linzen, T., Chrupala, G. & Alishahi, A. (eds.) *Proceedings of the Workshop: Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2018, Brussels, Belgium, November 1, 2018*, Association for Computational Linguistics, pp. 353–355, <https://doi.org/10.18653/v1/w18-5446>.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O. & Bowman, S. R., 2019b, 'GLUE: A multi-task benchmark and analysis platform for natural language understanding', *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, OpenReview.net.

Wang, Z., Zhang, J., Feng, J. & Chen, Z., 2014, 'Knowledge graph embedding by translating on hyperplanes', Brodley, C. E. & Stone, P. (eds.) *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada*, AAAI Press, pp. 1112–1119, <http://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/view/8531>.

Williams, A., Nangia, N. & Bowman, S. R., 2018, 'A broad-coverage challenge corpus for sentence understanding through inference', Walker, M. A., Ji, H. & Stent, A. (eds.) *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, Association for Computational Linguistics, pp. 1112–1122, <https://doi.org/10.18653/v1/n18-1101>.

Wu, Y., Gardner, M., Stenetorp, P. & Dasigi, P., 2022, 'Generating data to mitigate spurious correlations in natural language inference datasets', Muresan, S., Nakov, P. & Villavicencio, A. (eds.) *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, Association for Computational Linguistics, pp. 2660–2676, <`https://doi.org/10.18653/v1/2022.acl-long.190`>.

Yang, B., Yih, W., He, X., Gao, J. & Deng, L., 2015, 'Embedding entities and relations for learning and inference in knowledge bases', Bengio, Y. & LeCun, Y. (eds.) *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, <`http://arxiv.org/abs/1412.6575`>.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J. G., Salakhutdinov, R. & Le, Q. V., 2019, 'Xlnet: Generalized autoregressive pretraining for language understanding', Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B. & Garnett, R. (eds.) *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 5754–5764.

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', Riloff, E., Chiang, D., Hockenmaier, J. & Tsujii, J. (eds.) *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, Association for Computational Linguistics, pp. 2369–2380.

Zeng, S., Xu, R., Chang, B. & Li, L., 2020, 'Double graph based reasoning for document-level relation extraction', Webber, B., Cohn, T., He, Y. & Liu, Y. (eds.) *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, Association for Computational Linguistics, pp. 1630–1640, <https://doi.org/10.18653/v1/2020.emnlp-main.127>.

Zhang, F., Yuan, N. J., Lian, D., Xie, X. & Ma, W., 2016, ‘Collaborative knowledge base embedding for recommender systems’, Krishnapuram, B., Shah, M., Smola, A. J., Aggarwal, C. C., Shen, D. & Rastogi, R. (eds.) *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, ACM, pp. 353–362, <https://doi.org/10.1145/2939672.2939673>.

Zhang, S., Tay, Y., Yao, L. & Liu, Q., 2019a, ‘Quaternion knowledge graph embeddings’, Wallach, H. M., Larochelle, H., Beygelzimer, A., d’Alché-Buc, F., Fox, E. B. & Garnett, R. (eds.) *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pp. 2731–2741, <http://papers.nips.cc/paper/8541-quaternion-knowledge-graph-embeddings>.

Zhang, Y., Baldridge, J. & He, L., 2019b, ‘PAWS: paraphrase adversaries from word scrambling’, Burstein, J., Doran, C. & Solorio, T. (eds.) *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, Association for Computational Linguistics, pp. 1298–1308, <https://doi.org/10.18653/v1/n19-1131>.

Zhang, Z., Geng, X., Qin, T., Wu, Y. & Jiang, D., 2020, ‘Knowledge-aware procedural text understanding with multi-stage training’, *CoRR*, vol. abs/2009.13199, <https://arxiv.org/abs/2009.13199>.

Zheng, C. & Kordjamshidi, P., 2020, ‘SRLGRN: semantic role labeling graph reasoning network’, Webber, B., Cohn, T., He, Y. & Liu, Y. (eds.) *Proceedings of the*

2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, Association for Computational Linguistics, pp. 8881–8891, <https://doi.org/10.18653/v1/2020.emnlp-main.714>.

Zhong, W., Xu, J., Tang, D., Xu, Z., Duan, N., Zhou, M., Wang, J. & Yin, J., 2020, ‘Reasoning over semantic-level graph for fact checking’, Jurafsky, D., Chai, J., Schluter, N. & Tetreault, J. R. (eds.) *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, Association for Computational Linguistics, pp. 6170–6180, <https://doi.org/10.18653/v1/2020.acl-main.549>.

Zhou, B., Richardson, K., Ning, Q., Khot, T., Sabharwal, A. & Roth, D., 2021, ‘Temporal reasoning on implicit events from distant supervision’, Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tür, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T. & Zhou, Y. (eds.) *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, Association for Computational Linguistics, pp. 1361–1371.