Semantic, Syntactic and Joint Deep Learning of Event Extraction

Hao Anran

College of Computing and Data Science

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy (Ph.D)

2025

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

30/06/2024
Date

Hao Anran

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

20/07/2024

Date

Assoc Prof Hui Siu Cheung

Authorship Attribution Statement

This thesis contains material from 2 papers published in the following peer-reviewed journal and conference in which I am listed as an author.

Chapter 3 is published as Anran Hao, Anh Tuan Luu, Siu Cheung Hui, and Jian Su. "A Contrastive Learning Framework for Event Detection via Semantic Type Prototype Representation Modelling." *Neurocomputing*, 556:126613, 2023. ISSN 0925-2312. doi: https://doi-org.remotexs.ntu.edu.sg/10.1016/j.neucom.2023.126613.

The contributions of the co-authors are as follows:

- I came up with the key idea, designed all experiments, implemented all of the source code and conducted all experiments.
- I prepared the manuscript draft. The manuscripts were revised and edited together with Prof Luu Anh Tuan, Prof Hui Siu Cheung and Dr Su Jian.

and Anran Hao, Siu Cheung Hui, and Jian Su. "Semantic Pivoting Model for Effective Event Detection". *Proceedings of the 14th Asian Conference on Intelligent Information and Database Systems (ACIIDS)*, pages 534-546, Ho Chi Minh City, Vietnam, November 28-30 2022. doi: 10.1007/978-3-031-21967-2_43.

The contributions of the co-authors are as follows:

- I came up with the key idea, designed all experiments, implemented all of the source code and conducted all experiments.
- I prepared the manuscript draft. The manuscripts were revised and edited together with Prof Hui Siu Cheung and Dr Su Jian.
- Prof Luu Anh Tuan also assisted in the improvement of the manuscript draft.

30/06/2024
.....
Date

.....
Hao Anran

Acknowledgements

First and foremost, I would like to express my deepest gratitude to Dr. Su Jian and Prof. Hui Siu Cheung. Your unwavering support and warm encouragement throughout my PhD journey have been invaluable. I am deeply thankful for the countless times you have guided me with your insights and advice.

I would also like to thank Prof. Luu Anh Tuan, Dr. Tay Yi, and Dr. Sun Shuo. Your mentorship and guidance as senior researchers have been incredibly enriching and have greatly helped my growth as a researcher.

I am also profoundly grateful to the Agency for Science, Technology and Research (A*STAR), Singapore for sponsoring my PhD study. I would also like to express my sincere gratitude to Mr. Lau Chin Tiong Douglas and Dr. Tan Chee Wah Wesley for being the trustees of my scholarship. This research would not have been possible without your generous support.

To my lovely colleagues at the Institute for Infocomm Research (I2R), A*STAR and excellent peers from my graduate school, particularly Dr Chen Bin, Donovan Ong, Gao Yuze, Zhang Yuchen, Li Yue, Li Wenfeng, Liu Man, Li Zongmin, Yuan Haohan, Zheng Yandan, and Dong Xinshuai, thank you for the stimulating discussion. Your support and friendship have made this journey more enjoyable and fulfilling.

I would like to extend my heartfelt thanks to my best friend Wang Song, who is truly an angel who always understands and encourages me. Your unwavering love has been a constant source of strength. I am also grateful to my friends He Jingru, KC Huang, Li Rongshang, and many others for their friendship and encouragement.

Lastly, to my family, words cannot express my gratitude for your love, patience, and generosity. Your support has been the foundation of my achievements.

To all my friends

Contents

Acknowledgements **ix**

Abstract **xiii**

List of Figures **xix**

List of Tables **xxi**

Symbols and Acronyms **xxiii**

1 Introduction **1**
1.1 Motivation 1
1.2 Event Extraction Tasks 2
1.3 Objectives 4
1.4 Major Contributions 5
1.5 Organization of the Thesis 6

2 Literature Review **9**
2.1 Semantics-Based Methods for Event Detection 9
2.1.1 Event Detection 10
2.1.2 Label Representation Learning 11
2.1.3 Contrastive Learning 12
2.2 Syntax-Based Methods For Event Extraction 12
2.2.1 Sentence-Level Event Extraction 12
2.2.2 Document-Level Event Extraction 14
2.3 Joint Information Extraction For Event Extraction 15
2.3.1 Multi-task Learning 16
2.3.2 Dynamic Multi-Task Learning 16
2.4 Biomedical Event Extraction 19

3 Contrastive Learning Framework via Semantic Type Prototype Representation Modeling for Event Detection **23**
3.1 Background 23
3.2 Proposed Model 26

3.2.1 Unified Input-Label Encoding . . . 26
3.2.2 Contrastive Type Semantic Pivoting . . . 28
3.2.3 Trigger Classification . . . 30
3.2.4 Training . . . 30
3.3 Experiments . . . 31
3.3.1 Experimental Setup . . . 31
3.3.2 Experimental Results . . . 34
3.4 Case Studies . . . 39
3.5 Summary . . . 40

4 Soft Syntactic Reinforcement for Neural Event Extraction 41
4.1 Background . . . 41
4.2 Proposed Approach . . . 44
4.2.1 Intrinsic Syntactic Encoding . . . 44
4.2.2 Soft Syntactic Reinforcement . . . 45
4.2.3 Model Architecture . . . 47
4.3 Experiments for Sentence-level EE . . . 49
4.3.1 Experimental Setup . . . 49
4.3.2 Experimental Results . . . 52
4.4 Experiments for Document-Level EE . . . 53
4.4.1 Experimental Setup . . . 53
4.4.2 Experimental Results . . . 54
4.5 Case Studies . . . 56
4.6 Summary . . . 57

5 Dynamic Task Balancing for Joint Information Extraction 59
5.1 Background . . . 59
5.2 Problem Formulation . . . 62
5.3 Static Weighting Approach . . . 63
5.4 Proposed Method . . . 66
5.5 Experiments . . . 69
5.5.1 Experimental Setup . . . 69
5.5.2 Experimental Results . . . 72
5.5.3 Computational Cost Analysis . . . 75
5.6 Summary . . . 75

6 Biomedical Event Trigger Extraction 77
6.1 Background . . . 77
6.2 Proposed Model . . . 78
6.3 Experiments . . . 79
6.3.1 Experimental Setup . . . 79
6.3.2 Experimental Results . . . 81
6.4 Summary . . . 84

7 Conclusion and Future Work **85**
7.1 Summary 85
7.2 Future Work 87
7.2.1 Generative Methods for Event Extraction 87
7.2.2 Meta Learning for Joint Information Extraction 88
7.2.3 Few-Shot Event Extraction 89

List of Author's Awards, Publications and Submitted Works **91**

Bibliography **93**

Abstract

Much of human decision-making is based on a cognition of events. Events, ranging from simple occurrences to complex structures, form the essence of human communication, encapsulating rich information about actions, entities, and relationships. In the era of big data, empowering machines to recognize events in human language emerges as an important research problem for unlocking valuable insights from unstructured textual data. Event Extraction (EE), one of the main Information Extraction (IE) tasks in the field of Natural Language Processing (NLP), is immensely useful for real-world applications across different fields such as media, business, cybersecurity, and biomedical research.

Event extraction methods aims at automatically extracting event-related mentions from text documents written in natural language. However, traditional rule-based and statistical methods for event extraction often struggle to handle the inherent complexity and variability of natural language. This limitation has spurred significant interest in leveraging deep learning and neural network techniques to tackle the challenges of event extraction. Neural methods offer the promise of automatically learning intricate patterns and representations from data, enabling more effective and efficient event extraction systems.

Motivated by the growing importance of event extraction and the potential of neural approaches, this thesis aims to explore novel methods for enhancing event extraction from different aspects of deep learning. By delving into the nuances of event representation learning, semantic and syntactic understanding, and multi-task learning optimization, the thesis aims to push the boundaries of current event extraction systems and pave the way for more accurate and comprehensive event understanding.

In summary, the main contributions are: first, we propose the Contrastive Learning Framework via Semantic Type Prototype Representation Modeling for Event Detection (SemPRE), which utilizes pre-defined event type labels to capture event

type semantics. On two benchmark datasets, namely ACE 2005 and MAVEN, SemPRE achieves state-of-the-art results and demonstrates excellent performance in situations with limited training data, sentences containing multiple events, and trigger word disambiguation, thus offering a robust solution for more accurate and efficient event detection systems.

Second, we propose Syntactic Reinforcement for Neural Event Extraction (SRE) model for sentence-level and document-level event extraction. Sentence-level SRE outperforms other models on ACE 2005, CASIE, and PHEE, while document-level SRE outperforms prior state-of-the-art on MUC-4. It also achieves absolute improvements of 1.10%-11.81% in recall over the existing models on MUC-4. This is the first work that explores the feasibility of intrinsic syntactic mechanisms for event extraction, pushing the boundaries of extraction performance in both fine-grained sentence-level tasks and broader document-level event extraction.

Third, we propose the Adaptive Weighting Method for Joint Information Extraction (AWIE), a novel gradient-based optimization method to balance task losses for joint information extraction dynamically. On three multi-task IE datasets, AWIE outperforms the existing state-of-the-art Uncertainty and MGDA-UB techniques over all the tasks in F1 scores. For event extraction, AWIE improves the static-weighting DYGIE++ baseline by 1.7%-3.1% in F1 scores for the event trigger extraction and argument classification subtasks. It offers an effective strategy that dynamically balances task losses during joint IE model training.

Finally, we propose a domain-specific model for biomedical event trigger extraction, Bio-SemSyntEE, which incorporates the semantic-based mechanism from our proposed SemPRE model and the syntax-based mechanism from our proposed SRE model. Bio-SemSyntEE outperforms both discriminative and generative state-of-the-art models across three benchmark datasets. The results demonstrate the generalisability and robustness of our proposed mechanisms for domain-specific applications.

List of Figures

1.1 Overall organization of the thesis. 7

2.1 Decomposition of gradient search space of x for different task loss trade-offs. Each preference vector $\boldsymbol{u}_k$ represents a particular task loss trade-off preference. 18
2.2 The convergence behaviors of different dynamic weighting methods on the Pareto front. 19

3.1 Illustration of input sentence words that are trigger candidates and label words that are used to name pre-defined event types (in bold) sharing a unified word representation space. To enable the model to compare and contrast the different event types, we propose to *pivot* the ED model *semantically* based on the label words, which provide semantic meanings of the types, and leverage contrastive learning. . 24
3.2 Proposed SemPRE model architecture for event detection. 26
3.3 Performance results (%) of our SemPRE model in comparison with other ED models based on MAVEN according to various training data sizes. 36

4.1 Illustration of the traditional approach vs our proposed approach on integrating syntactic knowledge for event extraction. 42
4.2 Illustration of the dependency tree of a sentence with its corresponding syntactic depth vector and distance matrix. Our method relies on dependency parse trees, a classical structure to represent syntax. Take the sentence "*Last Monday, a 19-year-old extremist detonated a 30-kilo bomb near a military jeep, injuring three soldiers.*" as an example, its dependency parse tree can be converted to a syntactic depth vector or a syntactic distance matrix. 43
4.3 Proposed SRE model architecture for Sentence-level and Document-level Event Extraction. 45

5.1 An example of joint information extraction for entity extraction, relation extraction, and event extraction. 60
5.2 In multi-task IE, the learning of the tasks may require different scheduling. In this joint NER-RE model, the two tasks mature at different times. 64

5.3 Performance comparison of joint NER-RE models with different fixed loss weight ratios for individual tasks (single-task baselines are shown in dotted lines). As the static weight vs task performance does not show any pattern, it is hard to optimize in the way that it can benefit mainly a "main" task by adjusting the loss weight combination. 64
5.4 Model performance is often sensitive to the choice of loss weights. However, the performance results do not indicate the presence of a set of fixed loss weights that is optimal for all the tasks. Single-task NER and RE performances (dotted lines) are shown for reference. . 65
5.5 Model architecture for the joint IE experiments. 71

6.1 Bio-SemSyntEE model architecture. 78

List of Tables

3.1 Dataset split and statistics. 32
3.2 Performance results (%) for ED based on ACE 2005. The models using parsed syntactic or semantic features are marked with †, and the models using golden entity annotations are marked with ‡. . . . 34
3.3 Performance results (%) for ED based on MAVEN. The models using parsed syntactic or semantic features are marked with †, and the models using golden entity annotations are marked with ‡. T, S and G indicate **t**oken classifier, **s**equential decoder, and **g**raph-based decoder, respectively. 35
3.4 Performance results (%) of our SemPRE model in comparison with other ED models on single-event sentences (1/1) and multiple-event sentences (1/N) based on ACE 2005. 38
3.5 Performance results (%) on ACE 2005 and MAVEN in the ablation studies. 38
3.6 Case studies on ACE 2005. We show example sentences on the left, with the ground truth triggers underlined. In contrast to the ground truth types, we list the types that are predicted by the models, BERT+CRF, BERT_QA, CLEVE and SemPRE, on the right. We highlight the challenging triggers in **bold**, and the wrong model predictions with a × sign. 38

4.1 Detailed statistics for the sentence-level datasets, ACE 2005, CASIE and PHEE, including the number of documents, instances, events, and arguments, with average counts across 5 data splits. 49
4.2 Performance results (%) for Sentence-level Event Extraction. The best performance for each column is highlighted in bold and the second-best performance is underlined. 50
4.3 Performance results (%) on other syntax-based methods based on ACE 2005. 50
4.4 Performance results (%) of SRE on SSR with different PLMs based on ACE 2005. 50
4.5 Detailed statistics for the document-level dataset, MUC-4. 54
4.6 Performance results (%) for document-level Event Extraction based on Head Noun and Exact Match. 54
4.7 Performance results (%) for document-level Event Extraction based on CEAF-REE. 55

4.8 Performance results (%) of SRE with SSR and $\text{BERT}_{\text{Large}}$ on CEAF-REE. 55
4.9 Case studies for sentence-level and document-level EE between a SOTA model, the PLM baseline, and our proposed SRE model. For sentence-level EE, the triggers (Trg) are in bold with circled alphabetic labels and the arguments (Arg) are in italics with circled number labels. For document-level EE, sentence indices are in angle brackets. While triggers are not required, the argument candidates are in italics with circled number labels. '✓' indicates a correct prediction. 56

5.1 Statistics and data split of the datasets. 69
5.2 Performance results (%) of weighting methods on SciERC. NER denotes the entity extraction task, and RE denotes the relation extraction task. 72
5.3 Performance results (%) of weighting methods on ACE05-R. NER denotes the entity extraction task, and RE denotes the relation extraction task. 73
5.4 Performance results (%) of weighting methods on ACE05-E. NER denotes the entity extraction task, and RE denotes the relation extraction task. For event extraction, TE denotes the event trigger extraction task and AC denotes the argument classification task. . 73
5.5 Comparison of empirical computational time and memory costs of the methods on SciERC. 74

6.1 Dataset statistics of MLEE, GENIA 2009, and GENIA 2011. 80
6.2 Performance results (%) for Biomedical Event Trigger Extraction based on MLEE, GE09 and GE11. The best performance for each column is highlighted in bold. 82
6.3 Performance results (%) on the three datasets in the ablation studies. ΔF1 indicates the difference from the original model. 83

Symbols and Acronyms

Symbols

$\mathcal{R}^n$	the n-dimensional Euclidean space
$\|\cdot\|$	the 2-norm of a vector or matrix in Euclidean space
$\odot$	the Hadamard (component-wise) product
$\otimes$	the Kronecker product
$\langle\cdot,\cdot\rangle$	the inner product of two vectors
∇f	the gradient vector
$x_{i,k}$	the i-th component of a vector x at time k
$\bar{x}$	the vector with the average of all components of x as each element

Acronyms

NLP	Natural Language Processing
IE	Information Extraction
MTL	Multi-Task Learning
EE	Event Extraction
ED	Event Detection
EAE	Event Argument Extraction
NER	Named Entity Recognition
RE	Relation Extraction
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
CRF	Conditional Random Field

BERT	Bidirectional Encoder Representations from Transformers
BIO	Beginning-Inside-Outside Tagging Scheme
MGDA	Multiple-gradient Descent Algorithm
SemPRE	Semantic Prototype Representation Learning Framework for Event Detection
SSR	Soft Syntactic Reinforcement Mechanism
AWIE	Adaptive Weighting for Information Extraction
ACE	Automatic Content Extraction
MUC	Message Understanding Conference
MAVEN	Massive General Domain Event Detection Dataset

Chapter 1

Introduction

1.1 Motivation

In recent years, with the vast amount of unstructured textual information available on the web, there is a growing need for automated methods to extract relevant events and their associated information [1]. This extraction process is crucial for tasks such as information retrieval, question answering, summarization, and knowledge base population.

Event Extraction (EE) (specifically text-based EE) is the process of identifying and extracting structured information about events from unstructured text data [2]. An event typically consists of several components, including the trigger word that signifies the occurrence of the event and the participants involved (e.g., entities playing roles such as organization, device, or location). The challenge of EE lies in accurately identifying and linking these components together, especially in cases where events involve multiple participants or complex relationships [1, 3]. Effectively modeling the N-ary complex relationships requires sophisticated neural network architectures capable of capturing the contextual dependencies and semantic nuances present in the text data. To tackle the primary challenge of capturing the intricate N-ary relationships among the various elements of an event, we highlight the importance of improving the semantic and syntactic understanding capability of neural EE models to facilitate context understanding.

Furthermore, the importance of EE extends beyond academic research to real-world applications in various domains [3], including news analysis, social media monitoring, financial markets, biomedical science, and security. Automated EE systems can assist analysts and decision-makers in understanding and responding to rapidly evolving situations, facilitating timely and informed decision-making.

Through this thesis, we seek to contribute to the ongoing efforts to advance EE techniques by exploring and developing novel deep learning architectures and methodologies. Moreover, we extend the evaluation of our proposed models and methods to various application domains and test their generalizability and robustness. By addressing the challenges inherent in EE tasks, we aim to pave the way for more effective and reliable methods for extracting valuable knowledge from unstructured text data.

1.2 Event Extraction Tasks

Event extraction in deep NLP research refers to the task of automatically identifying and extracting structured information about events from natural language text. This task involves recognizing and categorizing different types of events along with their relevant attributes such as participants, time, location, and relationships.

EE can be further categorized based on the scope of input they consider. At the sentence level, EE [1, 2] focuses on extracting events and their associated information within individual sentences. This typically involves parsing each sentence to identify event triggers, such as verbs or nominalizations, and then determining the arguments or participants involved in the event, along with other relevant details.

On the other hand, document-level EE [4, 5] considers the broader context of multiple sentences or an entire document to extract events and their relationships across sentences. This involves not only identifying events within individual sentences but also implicitly resolving coreference, temporal relationships, and discourse coherence to capture the full narrative of events described within the document.

Both sentence-level and document-level event extraction have their unique challenges and applications. We formally formulate them as follows:

Sentence-level EE Given a sentence $S = \{w_1, \ldots, w_L\}$, where w_i refers to the i-th word token and L is the sentence length, the model outputs a sequence of labels $Y = \{y_1, \ldots, y_L\}$, where each y_i indicates all the event *triggers* (text spans that depict events) and the corresponding event *arguments* (text spans that denote entities taking certain roles in an event).

Document-level EE Given an input document comprised of d sentences $D = \{S_1, \ldots, S_d\}$, the document-level EE task aims to extract one or more structured events $Y = \{y_i\}$. Each event y_i has an event type t and a series of argument roles $(r_1^t, r_2^t, \ldots, r_N^t)$ to be filled by argument spans, where N is the number of argument roles associated with the type t. The event types and argument roles are from a set of pre-defined event types T and a set of role categories R. $t \in T$ and $\{r_1^t, r_2^t, \ldots, r_N^t\} \subseteq R$.

Traditional approaches to event extraction include rule-based systems and simple machine-learning methods that rely on handcrafted features. These methods are labor-intensive to develop and maintain and may not generalize well across different domains and languages [2, 6]. In contrast, deep learning methods have emerged as powerful techniques for automatically learning hierarchical representations of data, enabling more effective modeling of complex patterns.

The motivation behind this research stems from the desire to leverage the capabilities of deep learning techniques to advance the state-of-the-art in event extraction. By harnessing the power of neural networks, we aim to develop more accurate and robust models that can effectively capture the semantics and context of events from raw text. These models have the potential to significantly improve the performance of event extraction systems, enabling them to handle diverse types of events, languages, and domains with greater efficiency and scalability.

We identify three research gaps in deep learning event extraction. First, many existing models fail to fully leverage the semantic intricacies of language, leading to performance degradation, especially in low-resource settings. Second, despite the importance of syntactic structures in EE, current EE models based on pre-trained language models lack efficient mechanisms to exploit inherent syntactic bias and rely on external parsers, introducing error propagation and redundancy. Lastly, while joint learning methods show promise in improving generalization by learning multiple tasks simultaneously, they often suffer from suboptimal loss combination

strategies and lack dynamic mechanisms for task prioritization. These gaps hinder the development of more effective, robust, and adaptable EE models.

1.3 Objectives

The main objective of the research is to explore the linguistics and algorithmic techniques to address the current issues of deep learning event extraction models. In particular, our research will investigate techniques to advance the state-of-the-art methods for event extraction as follows:

- *Semantics-based Methods for Event Detection* - This aims to integrate label semantic information to improve representation learning for event detection. We will investigate a contrastive learning-based model for event detection so that the semantic information of the type labels can be integrated to facilitate representation learning.
- *Syntax-based Methods for Event Extraction* - This aims to enhance the implicitly embedded syntactic knowledge in pre-trained language models (PLMs). We will investigate a soft syntactic reinforcement mechanism that identifies syntax-related dimensions of PLM representation. In addition, we will investigate a novel syntactically enhanced neural model, which utilizes the soft syntactic reinforcement mechanism, for both sentence-level EE and document-level EE. On both levels and in various PLM settings, we will investigate the effectiveness of different variations of the proposed syntax-based mechanism over the performance of the EE baselines.
- *Joint Information Extraction for Event Extraction* - This aims to investigate the balancing of subtask losses in event extraction and multi-task IE, which contains multiple IE tasks related to event extraction. We will investigate existing dynamic weighting algorithms and their feasibilities for the joint IE problem. Moreover, we will also study the limitations of the current optimization approach of multi-task IE models, which mainly rely on static weighting techniques. Most importantly, we will investigate dynamic weighting methods and the development of a novel adaptive weighting algorithm for the tasks.

- *Domain-Specific Evaluation of the Methods on Biomedical Datasets* - This aims to study the effectiveness and generalisability of the proposed semantic-based and syntax-based methods for domain-specific applications. We will investigate a domain-specific model that combines a semantic-based and a syntax-based mechanism for biomedical event trigger extraction.

1.4 Major Contributions

As a result of this research, we have proposed several novel techniques for event extraction. Our main contributions are given as follows:

- *Contrastive Learning Framework via **Sem**antic Type **P**rototype **R**epresentation Modeling for **E**vent Detection (SemPRE)*: We have proposed SemPRE, a novel Transformer-based model that captures type label semantics for event detection (ED). SemPRE harnesses the words that are used to describe pre-defined event types to extract the semantics of these target types. SemPRE achieves remarkable F1 improvements of up to 11.2% over existing state-of-the-art ED models. Notably, SemPRE achieves these advancements without relying on additional annotated data or external linguistic resources. We have demonstrated the robustness of our model across various challenging scenarios, including situations with scarce data, multiple-event sentences, and uncommon trigger words.

- *Soft **S**yntactic **R**einforcement for Neural **E**vent Extraction (SRE)*: We have proposed a Soft Syntactic Reinforcement mechanism to enhance syntactic knowledge in pre-trained language models by enhancing syntactic knowledge intrinsically captured by PLMs. Experimental results on both sentence-level and document-level EE benchmark datasets have demonstrated the effectiveness of the proposed SRE model, which achieves state-of-the-art F1 results. For sentence-level EE, it surpasses existing syntax-based methods by over 2.08% and 2.00% absolute F1 on trigger and argument classification, respectively, and for document-level EE, it significantly improves recall by 3.81% (absolute measure). This contribution marks a significant advancement in leveraging syntactic information for event extraction tasks.

- ***A**daptive **W**eighting Method for Joint **I**nformation **E**xtraction (AWIE)*: We have identified the limitations of static weighting approaches commonly used in existing joint IE works. Moreover, we have explored the feasibility of dynamic weighting Multi-Task Learning (MTL) methods for joint IE. Specifically, we have proposed AWIE, a hybrid dynamic weighting method for joint IE. AWIE automatically balances tasks by assigning weights to their losses based on multi-objective gradient descent. Experimental results on three datasets have demonstrated the effectiveness of dynamic weighting methods for achieving superior results. Particularly, we have shown that AWIE outperforms existing baselines by dynamically assigning effective weights to task losses, thereby improving the overall performance. Moreover, AWIE offers flexibility in accommodating user task preferences by generating solutions with different task trade-offs.

- *Domain-specific model with **Bio**medical Pre-trained Encoder, **Sem**antic Pivoting and **Synt**actic Reinforcement for Biomedical **E**vent Trigger **E**xtraction (Bio-SemSyntEE)*: We have proposed Bio-SemSyntEE, a domain-specific model that integrates domain-specific pre-trained encoder with two mechanisms from our proposed SemPRE and SRE models to extract biomedical event triggers. Experimental results across three benchmark datasets show that Bio-SemSyntEE outperforms both generative and discriminative state-of-the-art models in precision, recall, and F1 scores. We have also shown that it has significant performance advantages over zero-shot and few-shot large language models. These results demonstrate that our proposed mechanisms are robust and generalizable to the biomedical domain.

1.5 Organization of the Thesis

Figure 1.1 illustrates the overall organization of the thesis. The main part consists of four technical chapters that present our major contributions of a semantic approach (SemPRE), a syntactic approach (SRE), a joint learning approach (AWIE) that explores event extraction jointly with other extraction tasks, and the integration of the first two in domain-specific (biomedical) event extraction. The four works are interlinked in that they are four canonical perspectives to advancing state-of-the-art deep learning for event extraction.

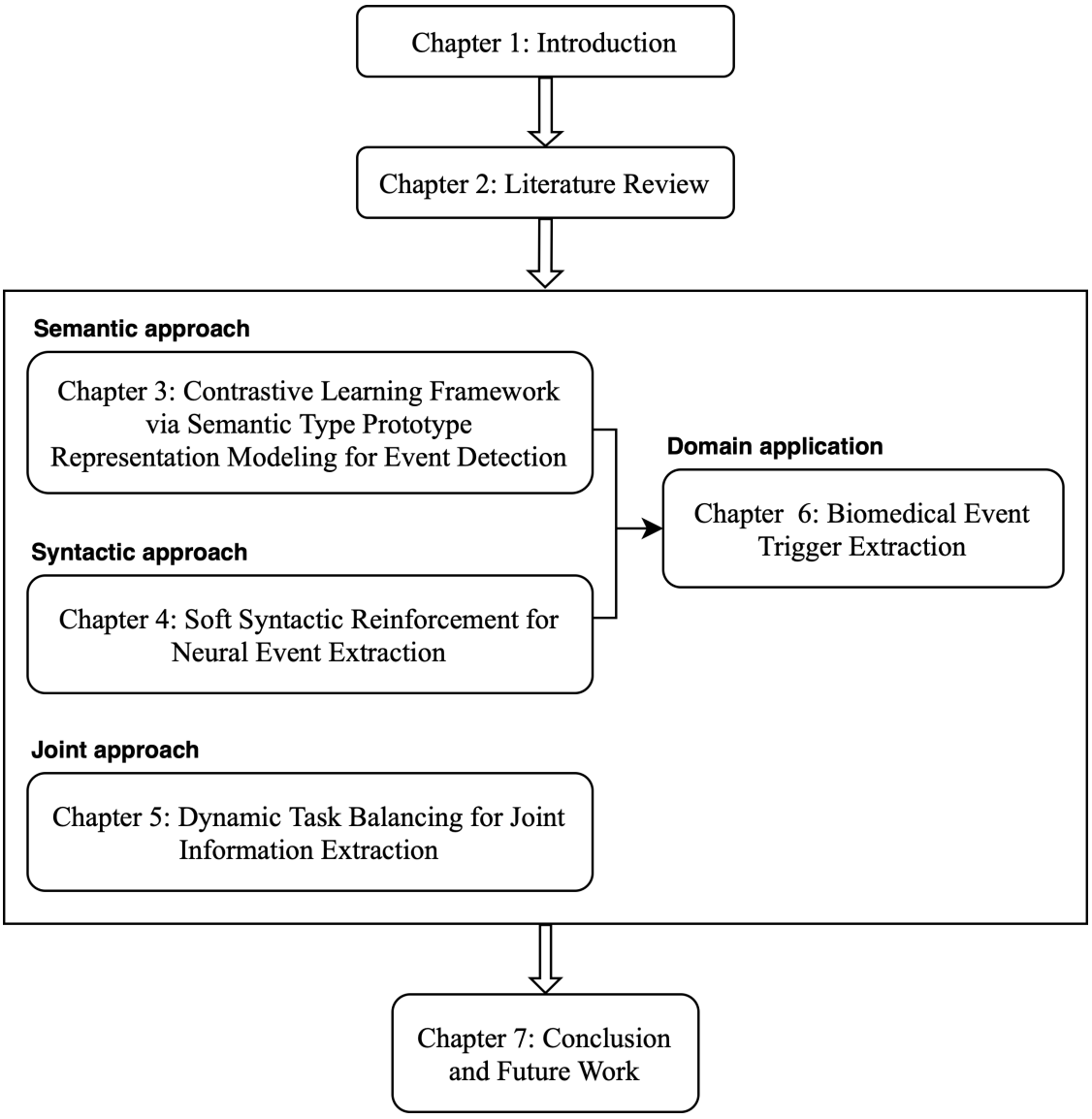

FIGURE 1.1: Overall organization of the thesis.

Chapter 1 provides an overview of the research problem and its background, and states the objectives and contributions of this thesis.

Chapter 2 reviews the existing literature and research in the domains of event extraction and joint information extraction (IE), highlighting key methodologies, challenges, and advancements.

Chapter 3 presents the first model in this thesis, SemPRE, a Contrastive Learning Framework via Semantic Type Prototype Representation Modeling for Event Detection. This chapter details the architecture of the proposed SemPRE model and its experimental evaluation. Experimental results on two benchmark datasets have

demonstrated its effectiveness in advancing the state-of-the-art in event detection its advantages in different scenarios.

Chapter 4 presents the Soft Syntactic Reinforcement (SSR) mechanism, a novel approach aiming at enhancing neural event extraction models syntactically. This chapter also introduces our proposed Soft Syntactic Reinforcement for Neural Event Extraction (SRE), which features the SSR mechanism. We present the empirical evaluation results of SRE on both sentence-level and document-level EE, demonstrating its significant performance improvements.

Chapter 5 explores dynamic weighting MTL methods for joint information extraction. This chapter presents the development of AWIE, a hybrid dynamic weighting method for joint IE tasks, and provides a thorough analysis of experimental results on three benchmark datasets, highlighting its effectiveness for multi-task IE compared to static weighting and existing dynamic weighting methods, and the advantage in accommodating user preferences.

Chapter 6 presents an application example on biomedical event extraction with our proposed SemPRE mechanism combined with domain-specific PLM and SSR-based syntactic reinforcement. This chapter describes the overall model architecture, implementation details, and performance results.

Chapter 7 concludes this thesis and presents the directions and challenges for future work.

Chapter 2

Literature Review

This chapter reviews the related work on event extraction. First, we focus on the task of event detection and review the work related to our work on SemPRE, a contrastive learning-based model that exploits label semantics. Next, we review syntax-based methods on sentence-level and document-level EE, which are related to our work on SRE, a syntax-based model that achieves the-state-of-art on both tasks. Then, we review the literature related to our work on AWIE, a dynamic weighting MTL optimization method for joint information extraction for EE. Finally, we review the related work on biomedical event extraction.

2.1 Semantics-Based Methods for Event Detection

Event detection is a subtask of event extraction that focuses on detecting whether an event of interest is mentioned in the text. The most widely adopted formulation of the task is given by the ACE Evaluation [7, 8], which defines event detection as the identification and classification of event triggers, which are words or phrases in a sentence that indicate the occurrence of an event. This process involves identifying the relevant words or phrases in a sentence that signify an event and categorizing them into pre-defined event types.

2.1.1 Event Detection

Traditional approaches to event detection relied heavily on handcrafted features and linguistic patterns, often requiring extensive domain knowledge and manual effort. Ahn [2] combined linguistic features including lexical and syntactic features, with external knowledge from WordNet, to build a feature-based model for extracting events. Ji and Grishman [9] advanced event extraction by merging global evidence from related documents with local decision processes. These traditional ED approaches relied heavily on handcrafted features and linguistic patterns, often requiring extensive domain knowledge and manual effort. In contrast, recent advances in deep learning have significantly enhanced event detection performance by automatically learning feature representations from large datasets [6, 10, 11].

Neural network-based models, starting from Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have demonstrated superior capability in capturing the complex dependencies and contextual information necessary for accurate event detection. Over the years, other than EE models based on CNN (e.g., DMCNN [6]) and RNN (e.g., JRNN [12]), more sophisticated architectures or mechanisms including attention [13], Transformer and graph neural network [14–16] have been proposed to improve the performance. Liu et al. [13] investigated a supervised attention mechanism to explicitly exploit argument information for ED. However, the problems of data scarcity and class imbalance remain the bottleneck for substantial improvement.

To alleviate the data scarcity problem, many works leverage external linguistic resources such as Freebase and Wikipedia [15, 17] for data augmentation. Utilization of pre-trained language models, joint extraction of triggers and arguments, and the incorporation of document-level information [18, 19] are also found to be able to enhance ED. In addition, approaches such as adversarial and reinforcement learning [15, 20], cross-lingual transfer learning [21, 22] and Δ-learning [23] have also been investigated to boost ED performance from different angles.

In our work, we assert that the semantic prior information about the pre-defined types is critical to the task. Current ED models are not informed of semantic information of the types. For example, Liu et al. [24] focused on trigger saliency attribution to address the issue of performance skewness, and proposed a new ED training mechanism that assesses the underlying patterns of event types. Liu et al.

[25] introduced a mechanism called dynamic prefix (DynPref) for dynamically adjusting the extraction process based on evolving context and query information. The DynPref mechanism constructs type-specific prefixes to integrate context information by learning a context-specific prefix for each context. Wang et al. [26] introduced a query-and-extract event extraction approach, which leverages different event annotations from various ontologies and the rich semantics in the queries to guide the EE process within a unified model. Recent works such as [25, 26] implicitly injected the type semantic information by including the label words as Transformer prompts. However, there is limited research on devising simple but effective mechanisms to explicitly perform type semantic learning.

2.1.2 Label Representation Learning

Class label representation has been used for image classification, but rarely explored for natural language processing (NLP) tasks. Among the few works, some encoded label information as system input for text classification [27–29]. Nguyen et al. [30] demonstrated the effectiveness of explicitly encoding relation and connective labels for discourse relation recognition. However, these methods learned separate encoders for the labels and the input sentence words. This is redundant because the words used in both the labels and the sentences are from the English vocabulary and they can share the same embedding. Furthermore, these methods do not effectively model the rich interactions between sentence words and labels as well as between two event labels. There is a lack of research encoding input text and label words with a unified scheme and using a deep attention-based structure to capture higher-order interactions within and between them.

Zhang et al. [31] and Huang and Ji [32] harnessed the power of pre-trained language models for zero-shot and semi-supervised ED. Both noticed the usefulness of type semantic information to the task, and yet neither directly utilized label words to conduct deep type representation learning. Zhang et al. [31] addressed zero-shot event extraction and used label words as seeds to manually curate "example trigger words" based on a large external corpus. Huang and Ji [32] proposed to learn prototypical type representations based on input sentences for both supervised and semi-supervised ED. However, this approach does not build a semantic linkage between label and input word representation. Despite the importance of event

type semantic information for ED, there is a gap in the literature, with few works deploying label-based type representation learning modules and exploiting pre-defined labels that carry this crucial semantic information.

2.1.3 Contrastive Learning

Contrastive learning is a discriminative technique in representation learning [33–35]. It is based on the principle of comparison between positive pairs of samples within a data class and negative pairs of samples from different data classes. Common contrastive training objectives include Contrastive Loss [34], Structured Loss [36], Triplet Margin Loss [37], and N-Pair Loss [38].

While several works have investigated contrastive learning for general, few-shot and zero-shot event extraction (e.g., [39–41]), the potential of the contrastive learning approach for modeling the event trigger prototypes has not been explored yet. More recently, Wang et al. [11] proposed a contrastive pre-learning framework, which leverages unlabelled data to enhance the pre-training with semantically parsed structure graph-based discrimination. However, this method only models the relationship between an anchor instance and its closest prototype. There is a gap in the current research in ED where novel type-oriented semantic contrastive losses are not being devised to learn the contrastive information of the relationships between prototypes and targets.

2.2 Syntax-Based Methods For Event Extraction

Event Extraction (EE) concerns more than just binary relations [42]. The current works on syntax-based methods for EE can be largely divided into two categories by granularity of the input: sentence-level EE and document-level EE.

2.2.1 Sentence-Level Event Extraction

For sentence-level EE, syntactic knowledge has always been relevant among the efforts that tackle the task. Early approaches to EE are feature-based methods (e.g., Li et al. [43], McClosky et al. [44]), in which syntactic features such as dependency

tree parsing plays a significant role. Li et al. [43] proposed a joint framework based on structured prediction that incorporates both local and global features such as dependency paths between triggers to capture dependencies among multiple triggers and arguments. With the development of pre-trained language models in NLP, EE systems use syntactic knowledge either as auxiliary information that can be added to word representation (e.g., [6, 12]) or as prior structures for learning graph neural network (e.g., [45, 46]). For example, Liu et al. [14] proposed the Jointly Multiple Events Extraction (JMEE) framework, which is an attention-based graph convolution network that features syntactic shortcut arcs.

Recent works largely fall into three categories: the pipelined approach, the joint approach, and the generative approach. TagPrime [47] was the latest pipeline model that uses priming words to enhance representation learning. On the other hand, DyGIE++ [48], OneIE [49] and AMR-IE [50] adopted a joint approach that learns multiple IE tasks in a unified manner, where EE benefits from learning to identify structures that are related to EE, such as named entities. DyGIE++ [48] utilized a span-based approach for joint information extraction, integrating BERT [51] for contextualized word representations and employing a multi-task learning framework with coreference propagation. OneIE [49] was a transformer-based unified joint extraction model that introduces a novel tagging scheme to jointly extract entities, relations, and events. AMR-IE [50] leverages Abstract Meaning Representation (AMR) graphs for event extraction, incorporating structural information from AMR graphs to capture semantic roles and event structures.

Generative methods, particularly those leveraging large pre-trained language models (PLMs), have garnered significant attention in the NLP community for their potential to improve performance across a wide range of tasks, including event extraction [52, 53]. For example, EEQA [18] employed a question-answering approach for event extraction, which utilizes PLM with a question generation module to extract events by posing natural language questions. RCEE [54] implemented a reinforcement learning-based framework for event extraction to iteratively refine event extraction results, while DEGREE-E2E [52] integrated graph neural networks (GNNs) in an end-to-end EE architecture to capture dependencies among events and entities within a document. However, the strength of generative methods mainly lies in low-resource settings such as open domain, zero-shot, and few-shot EE. Fully supervised discriminative methods under the pre-training + finetuning

paradigm still outperform generative methods and remain state-of-the-art [53, 55]. Moreover, Peng et al. [56] suggested that EE is a complicated *specification-heavy* task that LLMs may struggle with due to limitations in existing alignment methods and that dedicated instruction tuning is necessary for current generative baseline models to achieve reasonable performance through in-context learning.

In addition, while syntactic knowledge has long been important clues for event extraction [6, 12], probing studies on PLMs have found that syntactic structures can be emergent and intrinsic to language representation [57–60]. This indicates a potential alternative approach to the existing EE works. In our work, we propose a soft syntactic reinforcement mechanism to reinforce intrinsic syntactic knowledge learning.

2.2.2 Document-Level Event Extraction

Document-level event extraction, due to the scattering of argument entities, has been formulated as the task of converting the text containing event information into structured event description based on pre-defined templates, which was first defined as Template Filling in the MUC paradigm (MUC-4, 1992). State-of-the-art models include the following: MMR [61] was a multi-turn and multi-granularity, machine reading comprehension-based model. NST [4] proposed a sequence tagging model that uses a gate mechanism to merge sentence and paragraph representations. TempGen [62] treated the task as a template generation task, incorporating a copy mechanism that takes the top-k important cross-attentions as copy distributions into BART. RICB [5] leveraged Redundant Information and Closed Boundary Loss. The usage of syntactic knowledge in the existing work on sentence or document-level event extraction leverages additional representation learning on top of word embedding. In our work, we propose an approach to enhance the intrinsic syntactic knowledge in pre-trained language models.

2.3 Joint Information Extraction For Event Extraction

Event extraction consists of event trigger extraction and argument extraction. In the optimization process of learning an EE model, how to balance the two losses effectively remains a challenge. This can be viewed as multi-task learning. In fact, multi-task learning is an important approach for not just event extraction but other IE tasks such as named entity recognition and relation extraction. While there are task-specific systems that focus on a single task, joint IE systems offer a unified solution, reducing the need to build separate specialized models for multiple tasks with one system. Existing works for joint IE that are beneficial to EE are mainly based on structured prediction [43, 63], neural architectures [64], data augmentation [15] and unified MTL approaches [49, 65]. Li et al. [43] proposed a joint model for event extraction and coreference resolution which integrates features from both event extraction and coreference resolution tasks into a unified framework. Yang and Mitchell [63] introduced a joint neural network model for event extraction and temporal relation extraction, which focuses on jointly extracting events and temporal relations to capture the temporal ordering of events within a document. Nguyen and Nguyen [64] proposed Joint3EE, a joint event extraction and entity linking model that leverages mutual dependencies of the two tasks, simultaneously performing event extraction and entity linking to identify events and link them to relevant entities within the text. Wang et al. [15] proposed an adversarial training approach for event extraction, which is robust when handling noisy or adversarial input data. Luan et al. [65] proposed a general-purpose joint learning framework that integrates multiple NLP tasks, including event extraction, into a single unified model to leverage shared representations and dependencies. Lin et al. [49] introduced a joint extraction model based on a transformer architecture with a novel tagging scheme to jointly extract entities, relations, and events.

However, most existing multi-task IE models use static task loss weights [49, 66], which forbid any change in task weighting and often require an expensive hyper-parameter tuning process to achieve optimal performance [67]. In contrast to the architecture-based approach, we focus on the optimization-based improvement of the multi-task information extraction models, which allows for dynamic adjustment of task loss weights during the learning process.

2.3.1 Multi-task Learning

Multi-task Learning (MTL), which allows parameter sharing among multiple tasks, is a paradigm in machine learning that is popular in many domains including Natural Language Processing, Computer Vision, and Speech Processing [68]. For example in NLP, Dong et al. [69] presented a unified language model pre-training approach for both the understanding and generation tasks using multi-task learning, surpassing BERT and other state-of-the-art models across five natural language generation datasets. Learning the shared layers based on signals from different related tasks, MTL leads to more generalizable solutions for all the tasks. Existing MTL methods, depending on their strategy to balance tasks via loss weighting, can be classified into static weighting methods or dynamic weighting methods [70]. Static weighting methods pre-set weights that are fixed throughout the training process for task losses and combine them to form the overall objective function. The weights are usually either determined manually or found using extensive hyperparameter search (e.g., Grid Search and Random Search [71]). Grid search [71, 72] is a brute-force hyperparameter optimization technique that systematically searches through a pre-defined grid of hyperparameters. While it guarantees that all combinations within the search space covered by the grid are explored systematically, it is computationally expensive, especially for high-dimensional hyperparameter spaces, as it requires evaluating all possible combinations. Random search, in comparison, selects hyperparameters randomly from a pre-defined search space, and therefore it is more efficient than grid search, especially for high-dimensional search spaces. It does not guarantee that all combinations within the search space are explored but often discovers good hyperparameter settings faster than grid search. Both grid search and random search have the hyperparameters pre-defined before the training process begins, which does not allow the model to adapt to changing task demands, user task preferences or data characteristics. In contrast, the dynamic methods learn to assign adaptive weights to task losses in the training, thereby balancing the tasks dynamically.

2.3.2 Dynamic Multi-Task Learning

Dynamic weighting methods have been proposed over the years. Uncertainty-based weighting [67] and GradNorm [73] proposed to weigh task losses based on

uncertainty and gradient magnitudes for convolutional neural networks. Steepest descent algorithm for multi-objective optimization [74] was the first work to cast multi-task learning as multi-objective optimization, and they proposed an upper bound for the multi-objective loss. ParetoMTL [75] and its variation EPO search [76] proposed to compute the weights by searching for a set of well-distributed solutions on the Pareto front. These methods typically use feedback from the training process to update task weights or hyperparameters iteratively, allowing the model to adapt to changing task demands or data characteristics. They can potentially lead to better performance and faster convergence by allowing the model to adapt to the training data dynamically [75].

Specifically, we discuss three existing dynamic weighting methods in more detail, namely Uncertainty, MGDA-UB, and ParetoMTL. These methods have been developed to improve multi-task learning by dynamically adjusting task weights during training and are the most representative, widely applied methods among all.

Uncertainty [67] determines loss weights based on *task-dependent* or *homoscedastic* uncertainty, which is not dependent on the input data and varies between different tasks. For classification tasks like IE tasks, the joint loss is computed as follows:

$$L(\theta, \{\sigma_i\}) = \sum_{i=1}^{m} \frac{1}{\sigma_i^2} \mathcal{L}_i(\theta) + \log \sigma_i \tag{2.1}$$

where $\mathcal{L}_i$ is the cross entropy loss of task i and σ_i is the observation noise term of task i to be learned.

MGDA-UB [77] is a multi-objective optimization method based on the Multiple-Gradient Descent Algorithm (MGDA) [78]. The task weights are computed as follows:

$$\{w_i\} = \arg\min_{w} \left\{ \left\| \sum_{i=1}^{m} w_i \nabla_{G_i^{sh}} \hat{\mathcal{L}}_i(\theta) \right\|_2^2 \mid \sum_i w_i = 1, w_i \geq 0 \quad \forall i \right\} \tag{2.2}$$

where θ refers to both shared parameters and task-specific parameters, G_i^{sh} refers to the i-th task's gradient with respect to θ^{sh}, and $\hat{\mathcal{L}}_i$ denotes the empirical loss of task i. Note that the MGDA-UB algorithm proposes to optimize an upper bound under a non-singularity assumption that $\partial G_i^{sh} / \partial \theta^{sh}$ (the Jacobian of G_i^{sh}

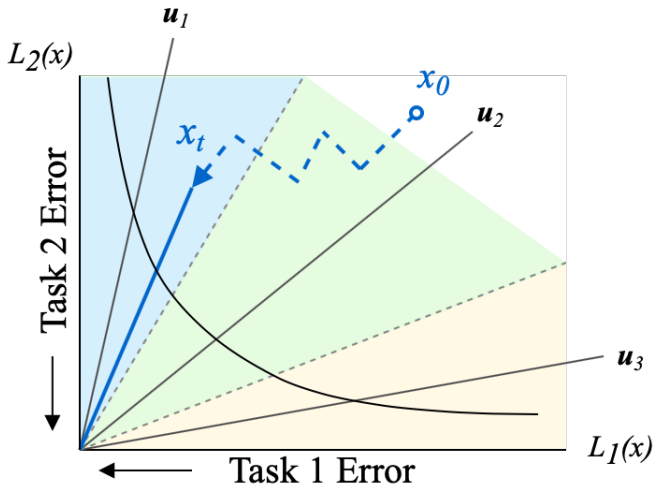

FIGURE 2.1: Decomposition of gradient search space of x for different task loss trade-offs. Each preference vector $\boldsymbol{u}_k$ represents a particular task loss trade-off preference.

with respect to θ^{sh}) is full-rank. This upper bound condition allows MGDA-UB to scale properly with the dimensionality of the gradients and the number of tasks.

ParetoMTL [75] augments the MGDA algorithm with problem space decomposition, extending the solution to a set of Pareto points with different task trade-offs. As shown in Figure 2.1, it does so by dividing the problem space into a few subproblems with a set of *preference vectors* $\boldsymbol{u}_k$. This forces the algorithm to find the Pareto solutions with each located in a different preference region, by encouraging gradient updates $x_t \rightarrow x_{t+1}$ to align with the selected preference vector. For example, in learning a multi-task model with learnable parameters denoted by a vector x, if a lower Task 1 loss is favored than a lower Task 2 loss, x falling in the blue region is most favorable. By posing a conditional constraint that the parameter vector x should form the smallest angle with u_1 among all $u_k, k \in \{1, 2, 3\}$, the algorithm reflects the task preference of Task 1 over Task 2. For the full ParetoMTL algorithm, please refer to Lin et al. [75].

However, our empirical experiments show that ParetoMTL has several issues that prevent its application to joint IE. First, it scales poorly and due to exploding computational complexity, it cannot be used to learn recent joint IE models, which are based on language models such as BERT and usually have parameter sizes more than 10^8 [51]. Second, as IE task losses are different in scale, the preference vectors that ParetoMTL originally proposed are no longer well-distributed in the

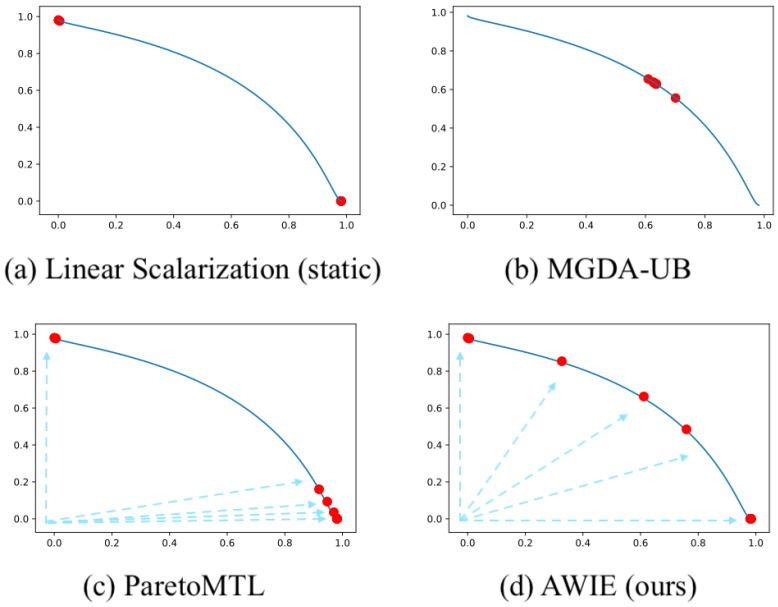

FIGURE 2.2: The convergence behaviors of different dynamic weighting methods on the Pareto front.

problem space. As shown in Figure 2.2[1], the dark blue line represents the Pareto front where Pareto solutions lie for a toy problem with a discrepancy in scales of task losses, and red dots illustrate the typical solutions a method can find. For Linear Scalarization, the static weighting method and MGDA-UB have no control over task trade-offs. ParetoMTL, using the preference vectors (light blue dotted arrows) to decompose the problem space, can find a set of solutions but they are rather biased against the task with a larger loss, resulting in a solution set that the trade-offs are similar to one another. Lastly, ParetoMTL proposes to generate evenly distributed unit vectors in the space, which makes the algorithm extremely time-consuming when the number of tasks is more than 2. For a detailed discussion on this limitation, please refer to Lin et al. [75].

2.4 Biomedical Event Extraction

Domain-specific event extraction is a specialized task within the broader field of information extraction that focuses on identifying and classifying events pertinent to a particular field, such as biomedicine, finance, or cybersecurity. Benchmark

[1]The Uncertainty method is not included because it does not solve for Pareto optimality.

datasets play a crucial role in the research process, providing standardized resources for training and evaluation. Notable examples in the biomedical domain include BioNLP'09 and BioNLP'11 GENIA datasets [79, 80], which focus on identifying various types of biological events such as gene expression, protein catabolism, and protein localization, and the Multi-Level Event Extraction (MLEE) dataset [81], which provides comprehensive annotations for various biomedical events relevant to cancer biology.

Biomedical event trigger extraction, which aims to detect the occurrence of biomedical events such as molecule binding and phosphorylation, is a canonical task in biomedical event extraction [79, 81]. While general-domain event trigger extraction deals with a wide variety of events across diverse contexts, biomedical event trigger extraction focuses on events that are important for biomedical applications and requires a nuanced understanding of specialized terminology, concepts, and relationships unique to the domain. This necessitates tailored approaches, such as domain-specific pre-training, where models are trained on domain-relevant corpora to better grasp the intricate contextual meanings within that field. For instance, in the biomedical domain, models might be pre-trained on scientific literature to effectively recognize complex biological interactions and events. One such model is SciBERT [82], which is pre-trained on a large corpus of scientific literature from the Semantic Scholar database. SciBERT leverages the structure and content of scientific texts to enhance its performance in extracting domain-specific events, demonstrating significant improvements over general-purpose models in biomedical NLP tasks.

Another notable example is BioBERT [83], which extends BERT by pre-training on PubMed abstracts and PMC full-text articles. BioBERT excels in various biomedical text mining tasks, showcasing its adeptness at handling the specialized language and dense information typical of biomedical literature. The state-of-the-art domain-specific pre-trained model for the biomedical domain is PubMedBERT [84]. Unlike earlier models such as BioBERT, which are based on mixed-domain pre-training, PubMedBERT pre-trains the model from scratch and achieves better performance across domain-specific fine-tuning tasks including information extraction.

Earlier biomedical event trigger detection works are feature-based machine learning models [85–89]. For example, Majumder [85] used extensive feature engineering to capture various linguistic and domain-specific characteristics and leveraged multiple event types and their interrelations. Turku Event Extraction System (TEES) [90] is a classical SVM-based model that uses a pipeline approach to extract biomedical entities, relations and events. In recent years, neural models that are based on representation learning and neural networks have become the predominant approach due to their superior performance [91–95]. Björne and Salakoski [91] expanded on their earlier work TEES by incorporating CNN, while Wang et al. [92] proposed BiLSTM-FastText, which utilized FastText embeddings [96] into a bidirectional long short-term memory model (BiLSTM). Wei et al. [95] introduced a multi-layer residual BiLSTM and relied on both word-level and character-level representation learning. Huang et al. [97] proposed GEANet, a BioBERT-based model with Hierarchical Knowledge Graphs to incorporate domain knowledge.

Chapter 3

Contrastive Learning Framework via Semantic Type Prototype Representation Modeling for Event Detection

To improve event detection representation learning via a semantic approach, we propose a Contrastive Learning Framework via Semantic Type Prototype Representation Modeling for Event Detection (SemPRE), which exploits the pre-defined event type labels to derive event type semantics via a unified input-label representation learning architecture. This chapter discusses the motivation and then presents the proposed model, experiments, and case studies.

3.1 Background

As introduced in Chapter 2, the state-of-the-art ED models are predominantly deep learning methods, which represent text using distributed real-valued vectors [1, 6]. Such approaches are more flexible and scalable than previous feature-based approaches [98, 99]. However, these neural models typically treat each event type class (e.g., DeclareBankruptcy, TransferOwnership, Attack, etc.) homogeneously as one-hot vectors and ignore the fact that the types are semantically meaningful. For example, as shown in Figure 3.1, the semantic information of the

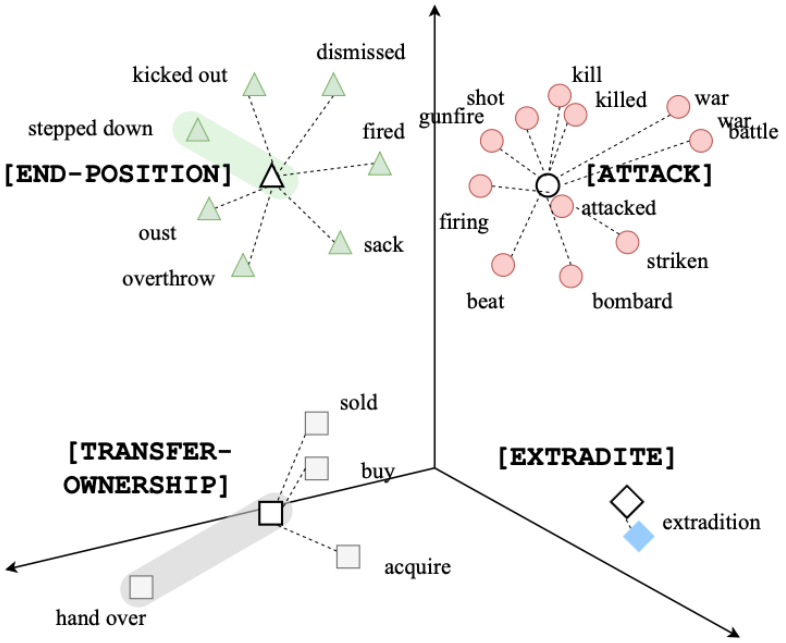

FIGURE 3.1: Illustration of input sentence words that are trigger candidates and label words that are used to name pre-defined event types (in bold) sharing a unified word representation space. To enable the model to compare and contrast the different event types, we propose to *pivot* the ED model *semantically* based on the label words, which provide semantic meanings of the types, and leverage contrastive learning.

target event types can be represented by the type label words, which share the word representation space with trigger candidate words in the input sentences.

Intuitively, ED models aim at learning to associate each candidate trigger word with an appropriate event type (if any). Existing models are agnostic to the semantic prior information about the types. This poses unnecessary difficulties for ED. Without knowing what each target type "means", the models have to infer type classification solely based on the training examples. This is inefficient and prone to overfitting, especially for unseen candidate words and rare types (such as TRANSFEROWNERSHIP and EXTRADITE in Figure 3.1), which have only limited instances. As discussed in the previous research works [3, 17, 100], data scarcity and the class imbalance problem of ED benchmarks have long been the bottleneck for the task [100].

Moreover, the relationship between types with associated or contrastive meanings was found relevant to ED [14, 101, 102]. Therefore, integrating a representation of

the meaning of each event type into the model enables it to exploit the correlation between the event types.

To overcome the above limitations, in this chapter we ascertain that the semantic meaning of the event types can be incorporated by using their class labels as input to guide ED. To this end, we propose a Contrastive Semantic Prototype Representation Learning Framework for Event Detection (SemPRE), which consists of three main modules. To use the event type label words to enhance ED, we first propose a Transformer-based *Unified Input-Label Encoding* module, which jointly encodes the input sentences and type labels, and enables input-label interaction attention. To synthesize type semantic representations with the representations of their instances, we design a *Contrastive Type Semantic Pivoting* module, which deploys a contrastive loss [33] and learns to discriminate input-type representation pairs. The *Trigger Classification* module generates type label prediction via sequence tagging for the input sentences.

The advantages of our proposed SemPRE model are three-fold. Firstly, it avoids the hassle of unsophisticated data augmentation methods, which may introduce much noise. Secondly, it does not involve any extra training data, which requires human efforts to gather and clean up. Thirdly, it does not use external syntactic or semantic parsing tools, which may have error propagation problems.

Overall, the main contributions of the work are summarized as follows: (1) We propose a model called SemPRE which exploits type label words via unified input-label representation learning for event detection. To the best of our knowledge, this is the first work to directly derive event type semantic information from label words for ED via contrastive learning. (2) We evaluate our SemPRE model on ACE 2005 and MAVEN benchmark datasets and achieve state-of-the-art performance. Without using external resources, our model outperforms several strong baselines using linguistic tools, and our results are comparable to previous models trained with extra data. (3) We show that our SemPRE model is also able to achieve robust performance under the scenarios of scarce training data and multiple-event sentences.

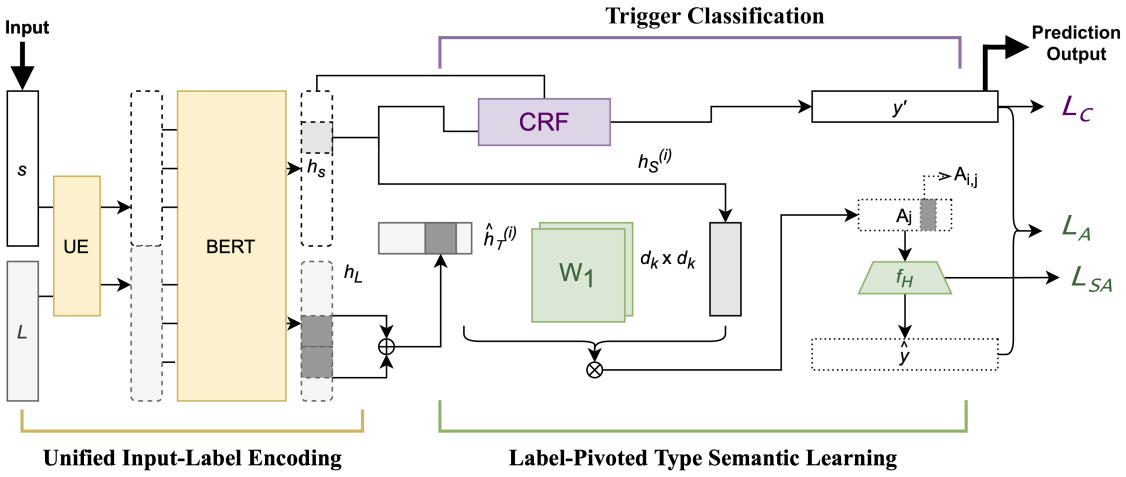

FIGURE 3.2: Proposed SEMPRE model architecture for event detection.

3.2 Proposed Model

In this section, we describe our proposed SemPRE model for event detection. Figure 3.2 shows the overall architecture of our proposed SemPRE model. The model consists of three modules:

- Unified Input-label Encoding - It is a BERT-based contextualized encoder that generates input and label representation in a unified manner.
- Contrastive Type Semantic Pivoting - It employs a contrastive-regularized loss to enrich input-label interaction and generate type semantic representations.
- Trigger Classification - It is a CRF-based decoder that generates predicted labels for input tokens.

SemPRE does not require any external knowledge base, linguistic tool or corpus to perform high-quality and robust ED. It mainly relies on learning implicit semantic association between input text and target type description (labels) to guide event detection.

3.2.1 Unified Input-Label Encoding

The model learns a unified encoding that applies to both the input sentence and the type labels. More specifically, we use the pre-trained BERT [51] embedder. We

concatenate each sentence with the type semantic sequence. After adding special tokens in BERT including `[CLS]` and `[SEP]`, for each input sentence s (of length N_S) with all the label words L (of length N_L), the input sequence is as follows:

$$X_{s,L} = \langle [\text{CLS}], L, [\text{SEP}_1], s, [\text{SEP}_2] \rangle \tag{3.1}$$

where `[CLS]` and `[SEP]` are special tokens used in BERT.

Note that L consists of the text names for describing concepts of the pre-defined event types, which are a part of the problem definition, and not the annotated output labels of the data. It consists of all the event types in the task definition and is fixed for all the training (and evaluation) instances. For example, there are 33 pre-defined event types in the ACE05 dataset in total, such as DeclareBankruptcy, TransferOwnership, and Attack. L is therefore a random sequence of the label words such as "transfer ownership declare bankruptcy ... attack". Given the input sentence "*The chain announced bankruptcy and was sold ... for £3 billion.*", the model constructs the input as "[CLS] transfer ownership declare bankruptcy ... attack [SEP] the chain announced ... billion [SEP]" according to Equation (3.1). In other words, in the training process, we use the same auxiliary information L, which is task-specific and not sentence-specific, for all input sentences to construct the input sequences.

As illustrated in Figure 3.2, the model jointly encodes the input sentence and the type labels using a unified embedding (UE). Subsequently, we obtain the BERT-encoded joint input-label representation sequence $(\mathbf{h}_L, \mathbf{h}_S)$:

$$(\mathbf{h}_L, \mathbf{h}_S) = BERT(X_{s,L}) \tag{3.2}$$

where $\mathbf{h}_L$ refers to the sequence of label word token representation and $\mathbf{h}_S$ refers to the sequence of input sentence token representation.

Following Devlin et al. [51], the attention heads in the BERT Transformer follow a unified form:

$$\text{Attention}(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{3.3}$$

where d_k is the hidden size and Q, K, V are the query, key, value matrices respectively. The attention heads enable the direct interaction between input sentences s and type label words L through multiple Transformer layers, resulting in enriched representations of both.

3.2.2 Contrastive Type Semantic Pivoting

In contrastive type semantic pivoting, we design a contrastive learning-based mechanism to learn type semantic representations. The basic idea is to move each type semantic representation closer to the candidate trigger tokens that belong to the type and farther from the tokens that do not. More specifically, the contrastive type semantic pivoting module treats a pair of input sentence tokens $s_i \in S$ and type $\mathcal{T}_j$, denoted by $(s_i, \mathcal{T}_j)$, as a positive sample if the correct event type for the token is $\mathcal{T}_j$. We use the input-type pairs of the remaining types $(s_i, \mathcal{T}_k)$, $k \neq j, k \in \mathcal{T}$, as the negative samples. In other words, by discriminating between positive and negative pairs, the module learns to generate type semantic representations that are (i) anchored by the label word representation and (ii) instantiated by candidate trigger word examples of the corresponding types from the training data.

For each type label in the pre-defined type set $\mathcal{T}$, we generate the type semantic representation by adding up the representation of all its tokens:

$$\hat{\mathbf{h}}_{\mathcal{T}}^{(i)} = \sum_{k=a}^{b} \mathbf{h}_{Lk} \tag{3.4}$$

where a and b are the start and end positions of the i-th type label $\mathcal{T}_i$ in the tokenized label sequence L. The length of $\hat{\mathbf{h}}_{\mathcal{T}}$ equals to the number of event types $|\mathcal{T}|$.

We use an interaction matrix $W_1 \in \mathbb{R}^{d_k \times d_k}$ to learn a similarity-based function between the two. For each input sentence token s_j and each type label $\mathcal{T}_j$, we have:

$$A_{i,j} = \sigma(\mathbf{h}_S^{(i)} W_1 \hat{\mathbf{h}}_{\mathcal{T}}^{(j)}) \tag{3.5}$$

where σ is the sigmoid nonlinearity and $A \in \mathbb{R}^{N_S \times N_{\mathcal{T}}}$ represents the input-label interaction. We then model a cross-type activation function H that maps A to the

final output prediction with a feed-forward network:

$$\hat{Y} = f_H(A) \tag{3.6}$$

We use the BIO tagging scheme [103] for the output labeling scheme. We denote the composition of Equations (3.5) and (3.6) as $g(.)$ and the contrastive loss is formulated as:

$$\begin{aligned} \mathcal{L}_A^{(i,j)} &= -g(\mathbf{h}_S^{(i)}, \hat{\mathbf{h}}_L^{(j)}) + \log\left(\sum_{k\in\mathcal{T}} \exp(g(\mathbf{h}_S^{(i)}, \hat{\mathbf{h}}_L^{(k)}))\right), \\ \mathcal{L}_A &= \sum_{i\in\mathcal{S}'}\sum_{j\in\mathcal{T}} \mathcal{L}_A^{(i,j)} \end{aligned} \tag{3.7}$$

where S' refers to the subset of S in which the candidate trigger tokens have positive labels. From another perspective, minimizing the contrastive loss in Equation (3.7) is equivalent to minimizing the KL Divergence between the predicted label distribution $\hat{Y}$ and the ground truth distribution Y:

$$\min\mathcal{L}_A = \min\mathcal{L}_{KL}(\hat{Y}, Y) \tag{3.8}$$

We sample the positive/negative pairs based on the labels of candidate trigger tokens. More specifically, we include all the pairs where the candidate trigger token has a non- "O" type label (i.e., the candidate token is indeed part of an event trigger word). The candidate trigger token and its corresponding type label form a positive pair, whereas the pairs containing the token and each of the other type labels are negative pairs.

Since not all type-type pairs have meaningful interaction, we add an L1 regularization term for the weight f_H (denoted by W_2) in Equation (3.6) to prevent the layer from overfitting. With this regularization loss, we force the weight W_2 to be sparse:

$$\mathcal{L}_{SA} = L1\_loss(W_2) \tag{3.9}$$

3.2.3 Trigger Classification

To optimize each output label prediction based on sentence-level label dependency, we use Conditional Random Fields (CRF) for decoding. The CRF [104] layer defines the probability of the label sequence $\mathbf{y}$ given the input sentence $\mathbf{x}$:

$$P(\mathbf{y}|\mathbf{x}) = \frac{\exp(score(\mathbf{x}, \mathbf{y}))}{\sum_{\mathbf{y}'} \exp(score(\mathbf{x}, \mathbf{y}'))} \tag{3.10}$$

$$score(\mathbf{x}, \mathbf{y}) = \sum_{i=0}^{n} A_{y_i, y_i+1} + \sum_{i=0}^{n} F_{\mathbf{x}, y_i} \tag{3.11}$$

where $\mathbf{A}$ is a transition matrix in which A_{y_i,y_i+1} is the transition parameter from the label y_i to the label y_{i+1}[1]. $\mathbf{F}_x$ is an emission matrix where F_{x,y_i} represents the scores of the label y_i at the i-th position. We decode the predicted event trigger labels from the predicted sequence with the highest score. We denote the cross-entropy prediction loss as $\mathcal{L}_C$.

3.2.4 Training

For the final objective function, we use a weighted linear surrogate loss function that is the combination of the cross-entropy prediction loss $\mathcal{L}_C$ from the trigger classifier, contrastive loss $\mathcal{L}_A$ from the type semantic pivoting module, and the regularization loss $\mathcal{L}_{SA}$ as in Equation (3.9):

$$\mathcal{L} = \mathcal{L}_C + \lambda * \mathcal{L}_A + \alpha * \mathcal{L}_{SA} \tag{3.12}$$

where $\lambda, \alpha \in C$ are weighting coefficients. The weighting coefficients are pre-defined and optimized with a random search scheme for the overall performance.

[1]The `start` and `end` labels are y_0 and y_{n+1} respectively.

3.3 Experiments

In this section, we first describe the datasets, evaluation metrics, implementation details, and compared models. We then present our experimental results and analysis.

3.3.1 Experimental Setup

Datasets and Evaluation Metrics We conduct the experiments based on the following publicly available ED datasets:

- ACE 2005 [2] - It is the most widely used benchmark dataset for event detection. The documents are from six different types of news sources and consist of event annotation for 8 event types and 33 more refined subtypes. We evaluate our models on its English subset. We use the same split as in the previous ED work [6, 48].
- MAVEN [100] - It is a recently published large-scale ED dataset containing 4,480 documents. It covers 168 event types organized into a hierarchical schema tree. The documents are based on Wikipedia entries and manually labeled with the help of a frame semantic parser. We use the official split and evaluation toolkit from [100].

The details of the datasets and splits are summarized in Table 3.1. Events refer to individual event mentions, i.e., per event trigger, rather than co-referential events. As negative trigger candidate boundaries are given in MAVEN but not in ACE 2005, the last column (i.e., #Negative) in Table 3.1 indicates the number of negative trigger candidates for MAVEN and negative sentences that do not include any event for ACE 2005 accordingly. For the evaluation metrics, we report precision (P), recall (R), and the micro-average F1 score. We use the micro-average F1 score instead of the macro-average F1 because it provides a more accurate reflection of performance across all classes, particularly in datasets with imbalanced distributions, which is typical in information extraction tasks like event detection[2].

[2]We use the micro-average F1 score for the same reason throughout the remainder of the thesis

TABLE 3.1: Dataset split and statistics.

Dataset	Split	# Doc	# Sent	# Event	# Negative
ACE 2005	Train	539	14,347	4,420	10,995
	Dev	30	634	505	341
	Test	40	840	424	493
MAVEN	Train	2,913	32,431	77,993	323,992
	Dev	710	8,042	18,904	79,699
	Test	857	9,400	21,835	93,570

Implementation Details We implement the proposed model in Pytorch [105]. To be consistent with previous works [100, 106], we use the Stanford CoreNLP toolkit for sentence segmentation. The non-content part of the raw text (e.g., XML tags) is excluded from the input. Except otherwise specified, we use the base uncased version of BERT. Maximum sequence length is set as 256 and input instances of size less than 256 are padded. We use the normal Xavier initialization. We use Adam [107] optimizer with the learning rate tuned around 3e-5 and warm-up steps between [0, 2000]. The batch size is set to 16 to be fit for single-GPU training. We implement early stopping (patience = 5) and limit the training to 50 epochs. We apply a dropout of 0.9 for both dense and attention layers. The values of λ and α are manually tuned between [1, 200] and [0, 1], respectively. For input construction as in Equation (3.1), we randomly shuffle the label name words from the pre-defined type set and apply the default tokenization to the sequence.

Compared Models We compare the performance of our model on ACE 2005 with four kinds of baselines: (1) architecture-based models, which are mainly based on vanilla neural architectures or their improved variants; (2) parsing-based models, which use linguistic tools to obtain auxiliary features; (3) resource-enhanced models, which use external corpora or annotated data; and (4) type representation learning models. The state-of-the-art models used for comparison are:

- DMCNN [6] - A canonical CNN-based model that proposes a dynamic multi-pooling mechanism.
- BERT_QA [18] - It performs ED through QA by constructing generic questions to query BERT.
- DMBERT [15] - It is a pipelined BERT-based model that adopts the dynamic multi-pooling mechanism in DMCNN.

- RoBERTa [11] - It uses the pre-trained RoBERTa language model instead of BERT.
- MOGANED [16] - It uses Multi-Order Graph Attention Network (GAT) to aggregate multi-order syntactic relations in the sentences based on Stanford CoreNLP parsed POS and syntactic dependency.
- GatedGCN [46] - It is a state-of-the-art graph-based model that uses a Graph Convolutional Network (GCN) with a gate diversity mechanism.
- DMBERT+Boot [15] - It is DMBERT trained on an augmented dataset *Boot* from an external corpus through adversarial training.
- DRMM-Image [108] - It is the state-of-the-art cross-modality model that constructs a news event image dataset for ACE 2005 to enhance ED.
- MLBiNet [109] - It is the state-of-the-art encoder-decoder model that uses a Multi-Layer Bidirectional Network to incorporate cross-sentence information to enhance sentence-level ED.
- CLEVE [11] - An ED model with enhanced task-specific pre-training on millions of news articles. It uses an encoder based on RoBERTa-large and a graph encoder based on AMR-parsed semantic structures.
- SS-VQ-VAE [32] - It is a BERT-based model that explicitly learns prototypical representation for event types. It filters candidate trigger words using an OntoNotes-based Word Sense Disambiguation tool and uses a Variational Auto-Encoder for regularization.

For MAVEN, we follow the various state-of-the-art baselines as established in the previous works [11, 100], among which four models (DMCNN, DMBERT, RoBERTa, and MOGANED) have been introduced above. The other models are as follows:

- BiLSTM - It is a vanilla recurrent bi-directional long short-term memory network for ED.
- BiLSTM+CRF - It is BiLSTM with Conditional Random Field (CRF) as the output layer to model structured dependencies.

TABLE 3.2: Performance results (%) for ED based on ACE 2005. The models using parsed syntactic or semantic features are marked with †, and the models using golden entity annotations are marked with ‡.

Model	Encoder	Decoder	P	R	F1
Architecture-based					
DMCNN†	Hybrid	Token classifier	75.6	63.6	69.1
DMBERT	BERT-base	Token classifier	77.6	71.8	74.6
BERT_QA	BERT-base	Token classifier	71.1	73.7	72.4
RoBERTa	RoBERTa-large	Graph-based decoder	75.1	79.2	77.1
Parsing-based					
MOGANED†‡	Hybrid	Token classifier	**79.5**	72.3	75.7
GatedGCN†	BERT-base	Token classifier	78.8	76.3	77.6
Resource-enhanced					
DMBERT+Boot	BERT-base	Token classifier	77.9	72.5	75.1
DRMM Image	BERT-base	Token classifier	77.9	74.8	76.3
MLBiNet‡	Hybrid	Sequential decoder	74.7	**83.0**	78.6
CLEVE†	Hybrid	Graph-based decoder	78.1	81.5	79.8
Type representation learning					
SS-VQ-VAE	BERT-large	Token classifier	75.7	77.8	76.7
SemPRE (ours)	BERT-base	Sequential decoder	77.9	82.9	**80.3**

- BERT+CRF - It is a vanilla BERT model that finetunes the pre-trained language model for ED and adopts CRF as the output layer.

3.3.2 Experimental Results

In this section, we discuss our experimental results, including comparison with state-of-the-art models, performance results of our proposed SemPRE model based on various data sizes and on single-event versus multiple-event sentences, ablation studies, and case studies.

Comparison with State-of-the-Art Models Table 3.2 shows the performance results based on the ACE 2005 dataset. We group the models by their approaches. Our proposed SemPRE model significantly outperforms the state-of-the-art models. With the same BERT-base language model, our SemPRE model achieves higher F1 than DMBERT and even RoBERTa (with 5.7% and 3.2% respectively). This shows the effectiveness of our proposed modules for ED. The parsing-based models, namely MOGANED and GatedGCN, achieve top precision scores, possibly due to

TABLE 3.3: Performance results (%) for ED based on MAVEN. The models using parsed syntactic or semantic features are marked with †, and the models using golden entity annotations are marked with ‡. T, S and G indicate **t**oken classifier, **s**equential decoder, and **g**raph-based decoder, respectively.

Model	**Encoder/Decoder**	**P**	**R**	**F1**
DMCNN	Hybrid/T	66.3	55.9	60.6
BiLSTM	BiLSTM/T	59.8	67.0	62.8
MOGANED†‡	Hybrid/T	63.4	64.1	63.8
BiLSTM+CRF	BiLSTM/S	63.4	64.8	64.1
DMBERT	BERT-base/T	62.7	72.3	67.1
BERT+CRF	BERT-base/S	65.0	70.9	67.8
RoBERTa	RoBERTa-large/G	64.3	72.2	68.0
CLEVE†	Hybrid/G	64.9	**72.6**	**68.5**
SemPRE (ours)	BERT-base/S	**67.3**	69.8	**68.5**

utilizing the parsed syntactic features. Without using any parsing tools, our proposed SemPRE model achieves a reasonably high 77.9% precision score. Moreover, SemPRE achieves 82.9% in recall, which is comparable to MLBiNet (with 83.0% in recall). As a result, SemPRE achieves the highest F1, which is 2.7%–4.6% higher than the parsing-based models.

Our proposed SemPRE model does not use any external resources but still achieves better results compared to resourced-enhanced models. This indicates the effectiveness of our proposed model and the importance of modeling type semantics. Lastly, we compare our SemPRE model with SS-VQ-VAE, which also performs type representation learning but does not use type labels. SemPRE achieves higher precision, recall and F1 (with +2.2%, +5.1% and +3.6% respectively) despite using a smaller BERT. This shows that exploiting the pre-defined labels for obtaining type semantics is a promising approach.

Table 3.3 shows the performance results based on MAVEN. We sort the models by their F1 performances in the table. SemPRE achieves 67.3% in precision and 68.5% in F1, which are both state-of-the-art performance. Similar to the evaluation on ACE 2005, the models with better encoders generally achieve better results on MAVEN. However, our proposed SemPRE model still outperforms the RoBERTa and DMBERT baselines in precision. Despite only using BERT-base as the encoder, SemPRE achieves high F1 which is comparable to CLEVE, which uses not only RoBERTa-large but also a graph encoder for structural pre-training. This confirms the effectiveness of our proposed modules including the CRF-based

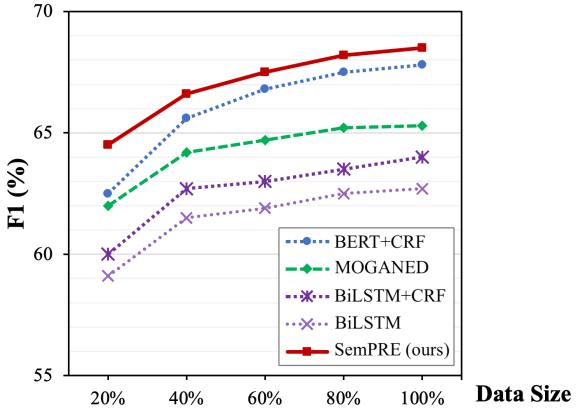

FIGURE 3.3: Performance results (%) of our SemPRE model in comparison with other ED models based on MAVEN according to various training data sizes.

trigger classifier. Our recall is lower (-2.8%) than the highest score achieved by CLEVE. This may be due to the reason that CLEVE uses substantial task-specific pre-training, and the large number of event types in MAVEN (4 times more than ACE) and their hierarchical relationships have added complexity for type representation learning.

Overall, our proposed SemPRE model achieves state-of-the-art performance on the ACE 2005 and MAVEN benchmark datasets. On both datasets, SemPRE has relatively higher recall than precision scores. Without using external resources, SemPRE matches the highly competent CLEVE and outperforms all the other baselines, demonstrating the effectiveness of our proposed modules. Further, we also highlight that although SemPRE does not rely on syntactic or semantic parsing tools, it achieves the best precision and reasonably high recall on the large MAVEN dataset.

Performance Based on Various Data Sizes We analyze the performance of our proposed SemPRE model in comparison with the baselines in the challenging scenario of data scarcity. We choose MAVEN for the experiments because it contains substantially more data and defines more event types than ACE 2005. Specifically, we randomly choose 20%, 40%, 60% and 80% of the samples from the training data for comparison with the baselines, which include BiLSTM, BiLSTM+CRF, MOGANED, BERT+CRF.

As shown in Figure 3.3, our proposed SemPRE model demonstrates robust performance across various training data sizes. Specifically, SemPRE achieves the

highest F1 among all the models on MAVEN with the training data size varying from 20% to 100%. With 80% of the training data, SemPRE outperforms (with 68.2% in F1) the baselines trained with 100% training data as shown in the figure. Its performance using 60% of the training data (with 67.5% in F1) still surpasses or matches other models' performances. In the extreme case where the training data is as scarce as only 20% of the original size, we observe that SemPRE still performs robustly (with 64.5% in F1), which is higher than all the other models. This demonstrates the effectiveness of our proposed model, especially the benefit of the unified input-label representation learning.

Performance Based on Single versus Multiple Event Sentences We analyze the performance of our proposed SemPRE model in terms of one/multiple-event sentences. We hypothesize that with the cross-type interaction enabled by the proposed modules, SemPRE should be able to learn to better model type relationships for the ED task and thereby performing well on both one-event and multiple-event sentences. Besides the state-of-the-art models DMCNN and MLBiNet, which are mentioned earlier, we compare our model with the strong baselines specifically designed to address this scenario. They are listed as follows:

- JRNN [12] - It is an RNN-based model with the memory features proposed for joint event extraction (i.e., ED and event argument extraction).
- JMEE [14] - It is a canonical baseline for multi-event ED that uses syntactic Graph Convolutional Network (GCN) to jointly extract events in sentences based on parsed dependency arcs.
- HBTNGMA [101] - It proposes a hierarchical and bias tagging mechanism to detect multiple events in one sentence collectively.

We choose to evaluate the models on the ACE 2005 benchmark dataset instead of MAVEN because the latter is only published recently and most existing works addressing this scenario have reported their performance based on ACE 2005.

Table 3.4 shows that our proposed SemPRE model achieves the state-of-the-art or comparative F1 performance on single-event and multiple-event sentences. Without using any linguistic parser or extra training data, our SemPRE model achieves

TABLE 3.4: Performance results (%) of our SemPRE model in comparison with other ED models on single-event sentences (1/1) and multiple-event sentences (1/N) based on ACE 2005.

Model	**1/1**	**1/N**	**all**
DMCNN	74.3	50.9	69.1
JRNN	75.6	64.8	69.3
HBTNGMA	78.4	59.5	73.3
JMEE	75.2	72.7	73.7
MLBiNet	**80.3**	77.4	78.6
CLEVE	75.5	82.7	79.9
SemPRE (ours)	78.9	**84.4**	**80.3**

TABLE 3.5: Performance results (%) on ACE 2005 and MAVEN in the ablation studies.

Model	**ACE 2005**		**MAVEN**	
	F1	**Δ F1**	**F1**	**Δ F1**
(1) full SemPRE	80.3	–	68.5	–
(2) without $\mathcal{L}_{SA}$	79.0	-1.3↓	68.1	-0.4↓
(3) without $\mathcal{L}_{SA}$+$\mathcal{L}_{A}$	77.6	-2.7↓	67.4	-1.1↓
(4) without $\mathcal{L}_{SA}$+$\mathcal{L}_{A}$+UILE	76.9	-3.4↓	67.2	-1.3↓

TABLE 3.6: Case studies on ACE 2005. We show example sentences on the left, with the ground truth triggers underlined. In contrast to the ground truth types, we list the types that are predicted by the models, BERT+CRF, BERT_QA, CLEVE and SemPRE, on the right. We highlight the challenging triggers in **bold**, and the wrong model predictions with a × sign.

#	**Sentence**	**Ground truth**	**BERT+CRF**	**BERT_QA**	**CLEVE**	**SemPRE**
(1)	Hoon said Saddam's regime was **crumbling** under the pressure of a huge air assault	EndOrg Attack	None × Attack	None × Attack	None × Attack	EndOrg Attack
(2)	Country A sent 1000 troops into country B, and the government said it would **send** more to prevent B rebels from **creating** an independent state.	Transport Transport StartOrg	Transport Transport None ×	Transport TranferMoney × None ×	Transport None × StartOrg	Transport Transport StartOrg

exceptionally high F1 (with 84.4%) performance on multiple-event sentences, significantly outperforming MLBiNet by 7.0%. It shows that our proposed SemPRE model can effectively model cross-event interaction, benefiting ED on multiple-event sentences. For single-event sentences, SemPRE achieves 78.9% in F1, which is higher than all the baselines except for MLBiNet. Moreover, we observe that SemPRE performs better at multiple-event sentences than at single-event sentences. One possible reason is that the interaction between multiple events helps event detection within a sentence, which is consistent with findings in previous works [14, 101, 102].

Ablation Study We conduct ablation experiments to show the effectiveness of the individual components of our model. Table 3.5 reports the results in F1 and their corresponding differences (Δ) with respect to the full-version SemPRE: (1) The original SemPRE model. (2) We remove the sparse regularization loss (denoted by "without $\mathcal{L}_{SA}$"), i.e., $\alpha = 0$. (3) We do not use the type semantic pivoting module at all (denoted by "without $\mathcal{L}_{SA}$+$\mathcal{L}_{A}$"), i.e., we only use the unified input-label encoding to inject type semantic information but does not explicitly learn label-based type representation. (4) On top of (3), we further remove the unified input-label encoding (UILE) module (denoted by "without $\mathcal{L}_{SA}$+$\mathcal{L}_{A}$+UILE").

The results show that all the key components in our proposed SemPRE model are necessary and effective for ED. Firstly, we observe that not using the regularization loss reduces performance by 1.3% in F1 for ACE 2005 and 0.4% for MAVEN. Secondly, removing the type semantic pivoting module leads to a significant drop in performance by 2.7% in F1 for ACE 2005 and 1.1% for MAVEN. This is possibly due to the reason that the type semantic pivoting module learns explicit type representations which are both label-based and context-aware.

Finally, we observe that performance degradation is significant if the event type labels are not used at all for type representation learning. Without the two proposed modules, the model performs 3.4% and 1.3% worse in F1 than the proposed SemPRE for ACE 2005 and MAVEN respectively.

3.4 Case Studies

As shown in Table 3.6, we conduct case studies and compare our SemPRE model with BERT+CRF and BERT_QA, which are similar to our model in nature, and CLEVE, which is the state-of-the-art model. BERT+CRF does not perform explicit type representation learning at all, while BERT_QA uses the word "verb" literally as an auxiliary query input (i.e., the whole input is "[CLS] verb [SEP] *sentence* [SEP]"). In sentence (1), "*crumbling*" is not a typical trigger word of an EndOrg event. None of the other models detect this trigger, whereas our proposed model can identify and classify it correctly. In sentence (2), our model successfully predicts both triggers. Firstly, the word "*send*" is synonymous with the first trigger "*sent*" in the sentence, which BERT_QA predicts its type as TranferMoney

and CLEVE misses. With type semantic learning, our SemPRE model can disambiguate "*send*" and classify it correctly as the TRANSPORT type. Secondly, while both BERT+CRF and BERT_QA miss the STARTORG event triggered by "*creating*", SemPRE correctly recognizes the trigger based on the semantic proximity of it and its context ("*an independent state*") to the type.

3.5 Summary

In this chapter, we have proposed a novel semantic learning model for ED, SemPRE, which exploits the pre-defined event type labels to derive event type semantics via a unified input-label representation learning architecture. Experimental results have shown that our proposed model significantly outperforms the state-of-the-art event detection models, without using extra annotated data or external linguistic resources. In addition, we have also demonstrated several other advantages of our proposed model over the baselines, such as working well for scenarios of scarce training data, multiple events in a sentence, and ambiguous trigger words.

Chapter 4

Soft Syntactic Reinforcement for Neural Event Extraction

In this chapter, we present our work on soft Syntactic Reinforcement for Neural Event Extraction (SRE), a novel model of syntactic approach to sentence-level EE and document-level EE. Based on the concepts of syntactic tree parse distance and depth, we introduce a Soft Syntactic Reinforcement (SSR) mechanism, which is the core component of SRE. This chapter discusses the motivation and then presents the proposed model, experiments and case studies.

4.1 Background

As discussed in Section 2 on Literature Review, the relevance of syntactic knowledge to event extraction (EE) has been recognized in various works. While recent EE systems gained performance advantages by leveraging large pre-trained language models, the syntactic reinforcement mechanism is achieved by embedding syntactic features separately from word representation. However, a line of probing studies (e.g., [57]) suggested that syntactic knowledge is *emergent* and *intrinsically* encoded in word representation learning. In this chapter, we present our work on Soft **S**yntactic **R**einforcement for Neural **E**vent Extraction (SRE), a novel neural model featuring syntactic reinforcement for EE, which benefits both sentence-level and document-level EE. We propose a soft syntactic reinforcement mechanism that

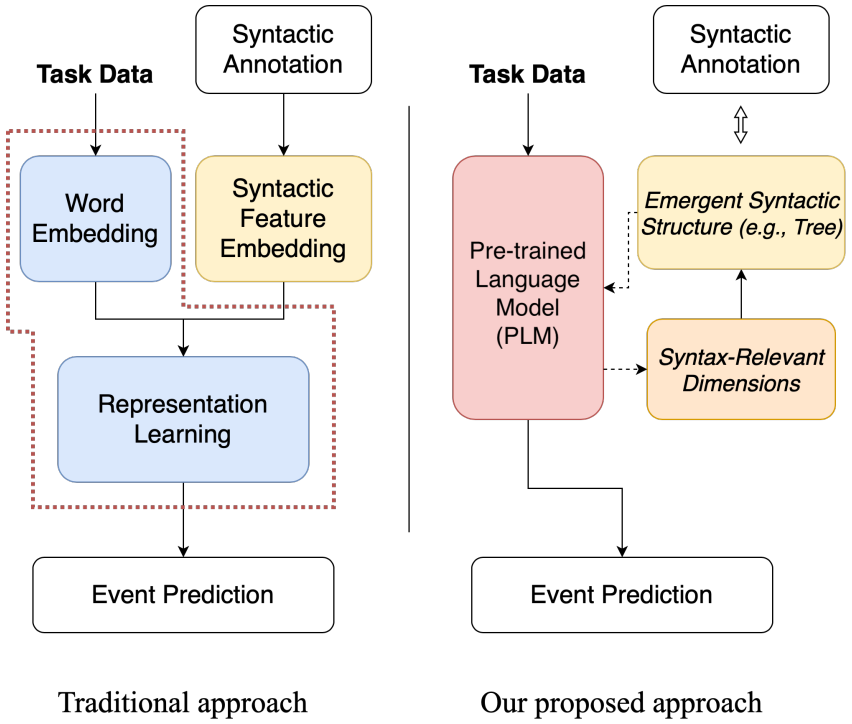

FIGURE 4.1: Illustration of the traditional approach vs our proposed approach on integrating syntactic knowledge for event extraction.

identifies syntax-related dimensions of PLM representation. It pre-trains a syntax-sensitive projection matrix that learns to recover syntactic trees. The EE model then uses the syntax-sensitive matrix for syntax-aware word representation learning. Our proposed method achieves state-of-the-art performance on both sentence-level and document-level EE benchmark datasets.

Among EE literature, syntactic information is found to be beneficial for the task [12, 16, 21, 46]. Syntactic features such as dependency trees provide clues for the models to better learn the interrelations between the candidate trigger words and their respective entities in sentences. With the advent of large pre-trained language models (PLMs), there has been a notable shift towards neural EE methods [42], which have demonstrated impressive performance gains. As shown in Figure 4.1, the traditional approach to EE relies on a "word embedding + representation learning" paradigm (red dotted lines), in which syntactic annotation is added to enhance the system via syntactic feature embedding. In other words, separate embeddings for words and syntactic features are trained and then combined for task-specific representation learning and prediction. In contrast, our proposed approach follows the latest EE works, which replace the "word embedding + representation learning" structure with a powerful PLM (the red block). With sufficient self-supervised

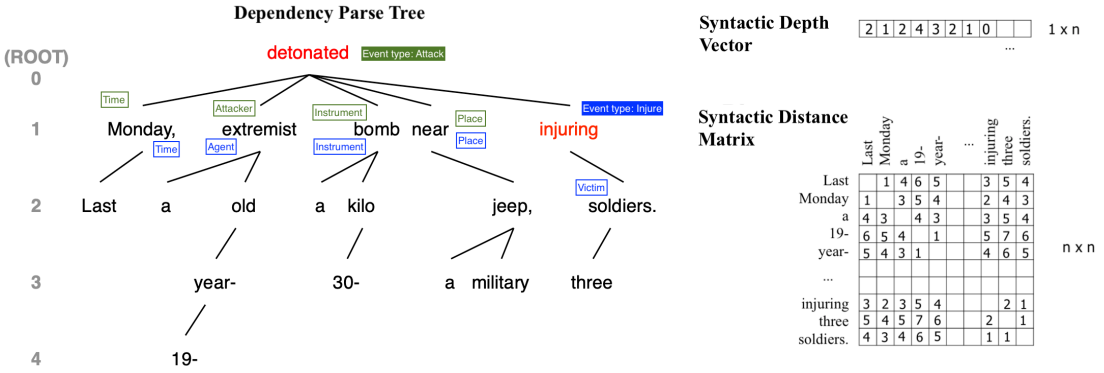

FIGURE 4.2: Illustration of the dependency tree of a sentence with its corresponding syntactic depth vector and distance matrix. Our method relies on dependency parse trees, a classical structure to represent syntax. Take the sentence "*Last Monday, a 19-year-old extremist detonated a 30-kilo bomb near a military jeep, injuring three soldiers.*" as an example, its dependency parse tree can be converted to a syntactic depth vector or a syntactic distance matrix.

pre-training, PLMs can significantly improve task-specific representation learning and lead to new state-of-the-art performance in EE.

Recent studies point out that the latest PLM-based EE models still make certain errors in event extraction due to their deficiency in syntactic structure knowledge [18, 49]. Similar to the traditional approach, research works such as Sachan et al. [110] attempted to inject syntactic knowledge into a PLM by learning external syntax-aware representations independently of the language model, and then combining them with the PLM. The experimental results have suggested that unless human-annotated parses on task data are used for training, the external representations are largely redundant and do not lead to performance gain for downstream NLP tasks. This aligns with the probing studies on PLM [57–60, 111], which found that syntax trees can be implicitly embedded in PLM's vector geometry, rather than being external to it.

In this chapter, we propose a novel approach to syntactically enhance neural EE models, which leverages a Soft Syntactic Reinforcement (SSR) mechanism. Without the need for syntactically annotating task data, our approach leverages a general-purpose syntactic resource, the Penn Tree Bank (PTB) dataset, for reinforcing the implicit syntactic knowledge learned by PLMs. More specifically, the SSR mechanism learns transformation matrices to activate syntax-relevant dimensions for emergent syntactic structures (including syntactic depth and distance) in a PLM and then uses the enhanced syntactic representations for event prediction.

To the best of our knowledge, this is the first work to explore this direction. Our results suggest that exploring syntactic enhancement mechanisms remains relevant for event extraction research and extends the study of innate syntactic structure in PLM.

Overall, the main contributions of this work are summarized as follows: (1) We propose a novel mechanism for injecting syntactic knowledge into neural EE models through a Soft Syntactic Reinforcement mechanism. This mechanism learns syntactic tree parsing in a distributed manner, providing a model with a deeper understanding of syntactic structures. (2) We propose the SSR-enhanced model, SRE, which features syntax-aware sentence representations for event extraction. (3) Experimental results on both sentence-level and document-level EE benchmark datasets show that our proposed SRE model achieves state-of-the-art performance in F1, with substantial improvements in recall for document-level EE tasks.

4.2 Proposed Approach

In this section, we first introduce the intuition of intrinsic syntactic encoding. Then, we describe the proposed Soft Syntactic Reinforcement mechanism, and the Syntax-Reinforced Event model that incorporates the SSR mechanism.

4.2.1 Intrinsic Syntactic Encoding

Studies on factual knowledge in PLM probing suggested that syntax trees can be embedded implicitly in deep models' vector geometry. In other words, deep contextual models encode the entire parse trees in their word representations. Hewitt and Manning [57] demonstrated that a low-rank transformation can recover the parse trees from BERT representations, without using any syntactically annotated data as input or being supervised to reconstruct them. Inspired by works along this line [57–60], we aim to enhance the syntactic knowledge that is captured in PLM representation learning layers by aligning a linear transformation of the relevant weights to a human-annotated syntactic tree dataset. Subsequently, we use the consolidated weights via supervised attention to enrich the task-specific learning.

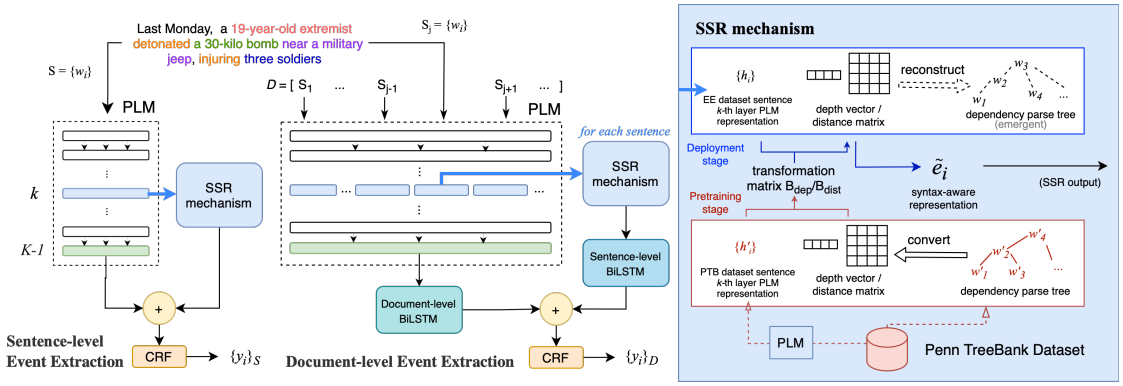

FIGURE 4.3: Proposed SRE model architecture for Sentence-level and Document-level Event Extraction.

We construct the emergent syntactic structure bias in the form of dependency parsing trees which can be decomposed into syntactic *depth* and *distance*. Figure 4.2 illustrates an example of dependency parsing based on depth and distance (event triggers are shown in red and argument role labels are shown in blue or green). **Syntactic depth** refers to the syntactic tree edge path length from each word to the root. For example, in Figure 4.2, the word "*detonated*" is the root, which has a syntactic depth of 0. The first word "*Last*", is two edges away from the root, and thus its syntactic depth is 2. The syntactic depth vector d of a n-word sentence has the dimension of n. **Syntactic distance** refers to the syntactic tree edge path length between each pair of words. For example, the word "*injuring*" has "*detonated*" as its parent. So the syntactic distance between "*injuring*" and "*detonated*" is 1. Similarly, the distance between "*injuring*" and its child "*soldiers*" is also 1. The syntactic distance matrix of a n-word sentence has the dimension of $n \times n$. Formally, given a syntactic tree T with the root node (word) r, the syntactic distance d_T measures the number of edges between each pair of nodes. The syntactic depth of node u is $d_T(u, r)$. For neighboring nodes u and v, the syntactic distance $d_T(u, v) = 1$.

4.2.2 Soft Syntactic Reinforcement

With the assumption of intrinsic syntactic encoding, we devise a soft syntactic reinforcement (SSR) mechanism to inject syntactic knowledge into the model. Unlike existing works, e.g., [45], we do not treat syntactic structures as external to word representation. Instead, we learn the transformation matrices that project PLM

representations to syntactical, new dimensional spaces, and use them to obtain soft syntactic representation of input tokens to reinforce the overall representation learning for the EE task.

Similar to the studies in Hewitt and Manning [57], the transformation matrix represents a linear transformation of a word representation space such that the transformed space embeds parse trees across all sentences. Let $\mathcal{M}$ be a model that takes in a sentence of n words $S = \{w_i\}, i \in \{1, \ldots, n\}$, and produces a sequence of vector representations $\{\mathsf{h}_i\}, i \in \{1, \ldots, n\}$. Given an annotated dependency parse tree of the sentence S, we define a transformation matrix B that maps h to approximate the depth vector $\{\mathsf{d}_T^S\} \in \mathbb{N}^n$ and the distance matrix $\{d_T^S\} \in \mathbb{N}^{n \times n}$. The transformation matrix B characterizes a syntax-salient normalization of the representation space learned by the language model.

We define the squared distances between the i-th and j-th words in the sentence S as:

$$d_B(\mathsf{h}_i, \mathsf{h}_j)^2 = (\mathsf{B}\mathsf{h}_i - \mathsf{h}_j)^\mathsf{T}(\mathsf{B}\mathsf{h}_i - \mathsf{h}_j) \tag{4.1}$$

and the squared depth norm of the i-th word as:

$$\|\mathsf{h}_i\|_B^2 = (\mathsf{B}\mathsf{h}_i)^\mathsf{T}(\mathsf{B}\mathsf{h}_i) \tag{4.2}$$

The loss functions for learning the transformation matrices for distance and depth are given in Equation (4.3) and Equation (4.4), respectively:

$$\min_B \sum_S \frac{1}{n^2} \sum_{i,j} |d_T(w_i, w_j) - d_B(\mathsf{h}_i, \mathsf{h}_j)^2| \tag{4.3}$$

$$\min_B \sum_S \frac{1}{n} \sum_i |\mathsf{d}_T(w_i)^2 - \|\mathsf{h}_i\|_B^2| \tag{4.4}$$

By minimizing the loss functions, the proposed mechanism learns the transformation matrices $\mathsf{B}_{distance}$ and B_{depth}. We use the Penn Tree Bank dataset, a manually annotated syntactic dependency dataset [112], for the pre-training stage.

4.2.3 Model Architecture

This section describes the architecture of our proposed soft Syntax-Reinforced Event (SRE) model, which integrates the SSR mechanism described in Section 4.2.2 into the baseline architecture. There are two considerations in choosing the baseline architecture. First, based on the recent literature and empirical experiments, we find that after carefully fine-tuning the PLMs, the baseline's performance on the task is comparable to that of the state-of-the-art models. Second, the selected baseline models should be relatively simple in terms of components, which alleviates the complex factors in analysis and benefits future study. For Sentence-level EE, we use the architecture which consists of a pre-trained PLM encoder, such as BERT [51] or RoBERTa [113], and a CRF decoder (Figure 4.3), as our baseline. For document-level EE, we adopt the Multi-Granularity Reader (MGR) model [4], which is the state-of-the-art extractive model, as the baseline.

We now formally describe how to integrate the baseline models with Soft Syntactic Reinforcement. Given a sentence $S = \{w_1, \ldots, w_L\}$, where w_i refers to the i-th word token and L refers to the sentence length, the model predicts $y_i, i \in \{1, \ldots, L\}$, as follows:

$$h_{1:L} = \mathsf{PLM}(w_{1:L}) \tag{4.5}$$

$$y_{1:L} = \mathsf{CRF}(h_{1:L}) \tag{4.6}$$

To incorporate the soft syntactic reinforcement, for example, using the depth-based matrix B^k_{depth} learned based on the k-th layer encoder representation, we obtain $\tilde{e}_i$, the linearly transformed word representations that contain syntactic tree information[1] as follows:

$$\tilde{e}_i = \mathsf{B}^k_{depth} h^k_i \tag{4.7}$$

[1]We call SSR "soft" because $\tilde{e}_i$ is obtained in a similar way as soft attention [114], i.e., by gathering the reinforced syntactic information across multiple dimensions, rather than selecting the "most syntactically relevant" dimension.

where h_i^k is the k-th layer PLM representation of the i-th word in a sentence[2]. For the distance-based SSR, B^k_{depth} is replaced by $\mathsf{B}^k_{distance}$ in this step. We then combine $\tilde{e}_i$ with the last-layer PLM representation h_i^{K-1} to obtain the fused representation r_i:

$$g_i = sigmoid(\mathsf{W_1} h_i^{K-1} + \mathsf{W_2}\tilde{e}_i + b) \tag{4.8}$$

$$r_i = g_i \odot h_i^{K-1} + (1 - g_i) \odot \tilde{e}_i \tag{4.9}$$

where K is the total number of PLM layers, $\odot$ represents element-wise product, and $\mathsf{W_1}, \mathsf{W_2}, b$ are learnable parameters. The fused representation $r_{1:L}$ with B_{depth} reinforcement is the input of CRF, replacing $h_{1:L}$ in Equation (4.6). For SRE with both distance and depth reinforcement, we fuse the distance-based $\tilde{e}_i$ and depth-based $\tilde{e}_i$ using the same gating function in Equations (4.8) and (4.9), and then fuse its outputs with h_i^{K-1}.

For document-level EE, as shown in Figure 4.3, we construct the model based on the MGR baseline architecture.

Overall, given a d-sentence document $D = \{S_1, \ldots, S_d\}$, for every sentence $S_j = \{w_1, \ldots, w_L\}, j \in \{1, \ldots, d\}$, where w_i refers to the i-th word token in D and L is the sentence length, the model first encodes each word token in the same way as Equation (4.5). Then, same as in Equation (4.7), the syntax-aware representation $\tilde{e}_i$ is generated via the SSR mechanism from h_i^k, which is the k-th layer PLM representation of the i-th word in a sentence. The syntactically reinforced representation of each sentence $\{\tilde{e}_i | w_i \in S_j\}$ is passed through a sentence-level BiLSTM to enrich contextual information across each sentence. Additionally, the sentence's last layer PLM outputs $\{h_i^{K-1} | w_i \in S_j\}$ are also passed through a document-level BiLSTM together with other sentences' PLM outputs, to infuse document context. Finally, similar to Equations (4.8) and (4.9), a gating function combines the BiLSTM outputs of each word with a learnable gating factor g_i, followed by a CRF to predict $\{y_i | w_i \in D\}$.

[2]The actual value of k is to be empirically determined based on the results from the development set.

TABLE 4.1: Detailed statistics for the sentence-level datasets, ACE 2005, CASIE and PHEE, including the number of documents, instances, events, and arguments, with average counts across 5 data splits.

Dataset	**Split**	**#Doc**	**#Inst**	**#Event**	**#Arg**
ACE 2005	Train	481	16,887	4,325	6,527
	Dev	59	1,957	503	792
	Test	59	2,076	520	778
	Total	599	20,920	5,348	8,097
CASIE	Train	701	1,044	5,973	15,890
	Dev	149	220	1,252	3,318
	Test	149	219	1,244	3,367
	Total	999	1,483	8,469	22,575
PHEE	Train	2,897	2,897	3,011	15,456
	Dev	965	965	1,002	5,117
	Test	965	965	1,006	5,186
	Total	4,827	4,827	5,019	25,760

4.3 Experiments for Sentence-level EE

This section describes our experimental setup, implementation details, and performance results on sentence-level EE.

4.3.1 Experimental Setup

Datasets We conduct performance evaluations on three benchmark datasets from the TextEE framework [115]: ACE 2005, CASIE and PHEE. For ACE 2005, as is introduced in Section 3.3.1, we evaluate all the models on its English subset. CASIE stands for Cyber Attack Sensing and Information Extraction and it is a cybersecurity domain event dataset that focuses on five event types: Databreach, Phishing, Ransom, Discover, and Patch. PHEE is a dataset for Pharmacovigilance Event Extraction. The annotations cover 5000 annotated events in medical case reports and biomedical literature. For all three datasets, we follow the TextEE framework [115] for data preprocessing and splits. The detailed statistics of the datasets are summarized in Table 4.1.

TABLE 4.2: Performance results (%) for Sentence-level Event Extraction. The best performance for each column is highlighted in bold and the second-best performance is underlined.

Model	ACE05			CASIE			PHEE		
	Tri-C	Arg-C	Arg-C+	Tri-C	Arg-C	Arg-C+	Tri-C	Arg-C	Arg-C+
DyGIE++	<u>71.31</u>	56.01	51.81	44.72	36.39	29.53	70.42	**60.84**	**45.65**
OneIE	71.05	59.93	**54.70**	70.57	54.23	22.05	69.98	37.51	29.76
AMR-IE	71.09	<u>60.62</u>	54.62	<u>71.83</u>	10.19	2.79	68.93	43.04	32.44
EEQA	70.04	55.28	50.36	42.79	35.14	26.23	70.29	40.40	32.02
RCEE	70.51	55.50	51.04	42.06	32.79	23.67	70.89	41.61	33.10
DEGREE-E2E	66.82	55.15	49.09	60.66	27.05	14.61	69.13	49.29	36.50
TagPrime	69.95	59.83	<u>54.64</u>	69.29	<u>61.03</u>	<u>49.07</u>	71.14	51.74	40.58
SRE (RoBERTa$_{\text{Large}}$)	**72.18**	**60.98**	54.45	**72.13**	**63.70**	**51.36**	**72.54**	<u>54.98</u>	<u>43.49</u>

TABLE 4.3: Performance results (%) on other syntax-based methods based on ACE 2005.

Model	**Tri-C**	**Arg-C**	**Arg-C+**
RoBERTa baseline	70.10	58.98	52.55
RoBERTa hard-synt	69.93	58.71	52.96
Syntax-RoBERTa	68.85	58.38	52.75
SRE (RoBERTa$_{\text{Large}}$)	**72.18**	**60.98**	**54.45**

TABLE 4.4: Performance results (%) of SRE on SSR with different PLMs based on ACE 2005.

PLM	**SSR**		**Tri-C**	**Arg-C**	**Arg-C+**
	dep.	dist.			
BERT$_{\text{Large}}$	-	-	66.43	53.14	48.61
BERT$_{\text{Large}}$	✓	-	69.13	55.53	49.96
BERT$_{\text{Large}}$	-	✓	68.23	55.05	49.74
BERT$_{\text{Large}}$	✓	✓	67.46	53.69	49.08
RoBERTa$_{\text{Large}}$	-	-	70.10	58.98	52.55
RoBERTa$_{\text{Large}}$	✓	-	**72.18**	**60.98**	**54.45**
RoBERTa$_{\text{Large}}$	-	✓	71.76	60.89	54.38
RoBERTa$_{\text{Large}}$	✓	✓	70.98	60.31	53.83

Evaluation Metrics We use micro-average F1 as the primary metric. Specifically, **Tri-C** scores a predicted trigger as correct if its offsets and type match a gold trigger. **Arg-C** scores a predicted argument as correct if its offsets, role, and event type match a gold argument. **Arg-C+** scores a predicted argument as correct if its offsets and role match a gold argument AND the corresponding predicted trigger's offsets and type match the gold trigger for the gold argument.

Implementation Details We implement the proposed model in Pytorch [116]. To be consistent with previous works [100, 106], we use the Stanford CoreNLP toolkit for sentence segmentation. The non-content part of the raw text (e.g., XML tags) is excluded from the input. In the experiments, we use the large cased version of BERT and the large cased version of RoBERTa as the baseline PLMs. From the experiments, we find the best k is 16 for both PLMs. Maximum sequence length is set as 256 and input instances of size less than 256 are padded. We use the normal Xavier initialization to set the initial random weights of the model. We use Adam [107] optimizer with the learning rate tuned around 3e-5 with a linear warm-up. The batch size is set to 24. We implement early stopping (patience = 5) and limit the training to 50 epochs. We apply a dropout of 0.9 for both dense and attention layers. The values of λ and α are manually tuned between [1, 200] and [0, 1], respectively. All experiments are trained on a single GPU and the best results are selected from 5 runs based on the evaluation results from the development set.

Compared Models We compare our proposed SRE model with the following three kinds of state-of-the-art models: (1) pipeline models, which include TagPrime [47]; (2) joint learning models, which include DyGIE++ [48], OneIE [49] and AMR-IE [50]; and (3) generation models, which include EEQA [18], RCEE [54], and **DEGREE-E2E** [52].

To assess the performance impacts of our proposed mechanism as a syntax-based enhancement approach, we compare it with the traditional hard syntactic approach (**RoBERTa hard-synt**) and a recent syntax-based mechanism called Syntax-RoBERTa [117]. For fair comparisons, our best SRE model and these two syntax-based models, as well as the PLM baseline, are all based on $\text{RoBERTa}_{\text{Large}}$. We implement the RoBERTa hard-synt approach by injecting SpaCy[118] dependency parsed features as learnable embeddings. We also conduct experiments to compare the performance of various SSR mechanisms on different PLMs. These two kinds of experiments are conducted based on ACE 2005, as it is the most commonly used sentence-level EE benchmark dataset.

4.3.2 Experimental Results

Performance Results Compared to State-of-the-art Models As shown in Table 4.2, in terms of F1, our proposed SRE model with the RoBERTa encoder consistently outperforms other models on ACE 2005 with its Tri-C F1 of 72.18% and Arg-C F1 of 60.98%. Its F1 performance for Arg-C+ is also comparable to the state-of-the-art models such as OneIE and TagPrime. On CASIE, we observe that our proposed SRE model performs better or is comparable to the current state-of-the-art models, with improvements of +0.30%, +2.67%, and +2.29% on F1 for Trig-C, Arg-C and Arg-C+ F1, respectively. On PHEE, SRE achieves the best F1 (72.54%) on Tri-C among all the models, and its argument classification outperforms all other models except DyGIE++. Overall, across all three datasets, we have shown that SRE achieves competitive results when compared with the state-of-the-art models, especially on the Tri-C task.

Performance of Syntax-based Methods In Table 4.3, we compare the performance of our proposed SRE model against RoBERTa hard-synt and Syntax-RoBERTa. The performance results show that our proposed SRE model based on RoBERTa outperforms the RoBERTa baseline with F1 improvement of +2.08%, +2.00% and +1.90% for Tri-C, Arg-C and Arg-C+, respectively. SRE also performs better than both RoBERTa hard-synt and Syntax-RoBERTa, with higher F1 across the subtasks of Tri-C, Arg-C and Arg-C+.

Performance of SRE on different PLMs and SSR As shown in Table 4.4, we evaluate the performance of SRE based on different PLMs with the SSR mechanism. For SRE with the RoBERTa encoder, the performance results show that the depth-based SSR achieves the best performance of 72.18% on F1, which is 2.08% higher than that without SSR. The distance-based SSR and depth+distance based SSR also improve the performance of SRE with RoBERTa, with F1 of 71.76% and 70.98% on Tri-C, respectively. For SRE with the BERT encoder, SRE with SSR also achieves improved performance results compared with that without SSR. On Trig-C, the depth-based, distance-based and depth+distance based SSR improve the performance on BERT over the BERT baseline by +2.70%, +1.80% and +1.03%, respectively. On Arg-C, the three variations of SSR lead to an F1 improvement of +2.39%, +1.91% and +0.55%, respectively. The improvements for

Arg-C+ are +1.35%, +1.07% and +0.47%, respectively, for the three variations. These results have demonstrated the effectiveness of our proposed SSR mechanism.

4.4 Experiments for Document-Level EE

In this section, we describe our experimental setup, implementation details and performance results on document-level EE.

4.4.1 Experimental Setup

Dataset For document-level EE, we conduct performance evaluation based on the MUC-4 benchmark dataset [119]. It consists of 1,700 newswire texts with 5 types of event templates about terrorist events. Each template specifies multiple arguments of concern to a type of event. We follow the prior work in data preprocessing and splits [5, 62]. Table 4.5 shows the statistics on the dataset and split details.

Evaluation Metrics Following previous works [4, 61, 120], we use micro-average Precision (P), Recall (R) and F1 based on Head Noun (HN) and Exact Match (EM) as the standard metrics. HN evaluates the predicted argument mentions only based on their head noun, whereas EM requires the predicted mentions to match the whole phrase of a gold one. To compare with more recent models, we report the results based on a new evaluation framework CEAF-REE [121], which encourages implicit coreference resolution.

Implementation Details We implement our proposed model based on PyTorch and Hugging Face's Transformer library. For each experiment we fine-tune BERT-large-cased. We find the best k is 16. The Bi-LSTM modules for both sentence-level and document-level contextualized learning have 3 layers each. We set the maximum sequence length as 200 for sentences and 512 for paragraphs. Each model is trained on a single GPU. The average training run-time for each model is 3.5 hours. The models are trained for 15 epochs with a batch size of 5. We use SGD as the optimizer with a learning rate decay. We implement early stopping with

TABLE 4.5: Detailed statistics for the document-level dataset, MUC-4.

Split	#Doc	#Sent	#Arg
Train	1,300	18,967	2,551
Dev	200	3,112	483
Test	200	2,786	533
Total	1,700	24,865	3,567

TABLE 4.6: Performance results (%) for document-level Event Extraction based on Head Noun and Exact Match.

Model	Head Noun			Exact Match		
	P	R	F1	P	R	F1
Cohesion	57.80	59.40	58.59	-	-	-
NST	56.44	62.77	59.44	52.03	56.81	54.32
MMR	**63.95**	58.71	61.19	**60.66**	55.34	57.87
SRE ($\text{BERT}_{\text{Large}}$)	61.93	**67.75**	**64.71**	56.84	**63.10**	**59.81**

the patience of 3. The learning rates ranging from [1.5e-4, 1.5e-2] are evaluated, of which the starting learning rate of 1.5e-3 has the best performance on the validation set.

Compared Models We compare our proposed SRE model with the following state-of-the-art models: Cohesion [120], MMR [61], and NST [4] on the standard evaluation metrics, and TempGen [62] and RICB [5] on the new CEAF-REE metric.

4.4.2 Experimental Results

State-of-the-art Results Table 4.6 reports the performance comparison of our proposed SRE model with the state-of-the-art models on MUC-4 using the standard evaluation metrics. We observe that SRE achieves the highest HN F1 with 64.71%, surpassing MMR by +3.52%. It achieves 59.81% in EM F1, which is 2.06% higher than MMR and 5.49% higher than NST. For both Head Noun (HN) and Exact Match (EM), our model significantly improves the recall compared to other models. More specifically, our proposed SRE model achieves 67.75% in HN recall, which is 8.35% higher than the feature-based Cohesion model, and 4.98% and 9.04% higher than the neural models NST and MMR, respectively.

TABLE 4.7: Performance results (%) for document-level Event Extraction based on CEAF-REE.

Model	**P**	**R**	**F1**
Cohesion	58.38	39.53	47.14
NST	56.82	48.92	52.58
GRIT	64.19	47.36	54.50
TempGen	**68.55**	49.90	57.76
RICB	57.68	58.03	57.85
SRE ($\text{BERT}_{\text{Large}}$)	55.93	**61.84**	**58.95**

TABLE 4.8: Performance results (%) of SRE with SSR and $\text{BERT}_{\text{Large}}$ on CEAF-REE.

PLM	**SSR**		**F1**	**Δ F1**
	dep.	dist.		
$\text{BERT}_{\text{Large}}$	-	-	57.34	-
$\text{BERT}_{\text{Large}}$	✓	✓	58.95	+1.61%
$\text{BERT}_{\text{Large}}$	✓	-	58.74	+1.40%
$\text{BERT}_{\text{Large}}$	-	✓	58.55	+1.21%

On CEAF-REE, as shown in Table 4.7, our proposed SRE model achieves 58.95%, which is the highest F1 among all the compared models. It is +1.10% higher than RICB, the second-best model. It also achieves exceptionally high recall, 61.84%, which is +3.79% higher than RICB.

Comparison with PLM baseline As shown in Table 4.8, we compare the performance based on the SSR mechanism for our SRE model for document-level EE. We found that SRE with depth+distance based SSR achieves the highest F1 of 58.95%, which is +1.61% higher than the $\text{BERT}_{\text{Large}}$ baseline. SRE with the depth-based SSR mechanism obtains F1 of 58.74%, whereas SRE with the distance-based SSR mechanism obtains F1 of 58.55%. The performance improvements over the baseline are +1.40% and +1.21%, respectively. Generally, SREs with the three variations of the SSR mechanisms have similar improved performance performance for document-level EE.

TABLE 4.9: Case studies for sentence-level and document-level EE between a SOTA model, the PLM baseline, and our proposed SRE model. For sentence-level EE, the triggers (Trg) are in bold with circled alphabetic labels and the arguments (Arg) are in italics with circled number labels. For document-level EE, sentence indices are in angle brackets. While triggers are not required, the argument candidates are in italics with circled number labels. '✓' indicates a correct prediction.

Sentence-level EE	**Gold**	**OneIE**	**RoBERTa$_{\text{Large}}$**	**SRE**
Eight people ①, including *a pregnant woman* ② and *a 13-year-old child* ③ were **killed** ⓐ in Monday's Gaza raid.	ⓐTrg[ATTACK]	✓	✓	✓
	①Arg[TARGET]	✓	✓	✓
	②Arg[TARGET]	✓	✓	✓
	③Arg[TARGET]	NONE	NONE	✓
It does rather delightfully remind me of *Coca-Cola* ① basically giving away *the bottling franchise* ② and then spending billions to **buy** ⓐ it back.	ⓐTrg[TRANOWN]	✓	✓	✓
	①Arg[BUYER]	NONE	NONE	✓
	②Arg[ARTIFACT]	✓	✓	✓
Document-level EE	**Gold**	**RICB**	**BERT$_{\text{Large}}$**	**SRE**
... ⟨2⟩ The massacre against the *Salvadoran Workers National Union Federation* ① (*FENASTRAS*) ② was planned in cold blood. ... ⟨4⟩ We have trustworthy information from our intelligence organs that this action was ordered by *Colonel Ponce* ③, that *Cristiani* ④ knew about it and approved it, and that it was carried out by *Colonel Elena Fuente* ⑤ as the head of the morbid death squad ... ⟨7⟩ Only a few days ago, *ARENA* ⑥ *assassins* ⑦ tried to kill the *president of the Mortgage Bank*⑧, *Mr Mason* ⑨, for not following their orders. ...	ATTACK Event 1			
	①TARGET	✓	✓	✓
	②TARGET	✓	NONE	✓
	③PERPIND	✓	✓	✓
	④PERPIND	NONE	NONE	✓
	⑤PERPIND	NONE	NONE	✓
	⑥PERPORG	✓	✓	✓
	ATTACK Event 2			
	⑦PERPIND	✓	✓	✓
	⑧NONE	✓	✓	TARGET
	⑨TARGET	NONE	NONE	✓

4.5 Case Studies

To better understand how our proposed SSR mechanism helps EE, we present three case studies based on the ACE 2005 and the MUC-4 development set, which are shown in Table 4.9[3]. For sentence-level EE, the sentence "*Eight people, including a pregnant woman and a 13-year-old child were killed in Monday's Gaza raid*" has the event trigger word "*killed*" (event type= CONFLICT:ATTACK) with the three corresponding arguments, which are in italics. While OneIE and RoBERTa$_{\text{Large}}$ fail to identify the third argument mention "*a 13-year-old child*" as a TARGET, our proposed SRE model can correctly identify it. We found that the predicted

[3]TRANOWN, PERPIND and PERPORG are short forms of TRANSFER-OWNERSHIP, PERPETRATOR INDIVIDUAL and PERPETRATOR ORGANIZATION, respectively.

parse depths of the head words of the two argument mentions, "*woman*" and "*child*", are similar to one another, possibly providing a clue for the task. The second example for sentence-level EE has a trigger word "*buy*", which is of the event type TRANSACTION:TRANSFER-OWNERSHIP, and two argument mentions, "*Coca-Cola*" (role=BUYER) and "*franchise*" (role=ARTIFACT). While OneIE and $\text{RoBERTa}_{\text{Large}}$ can only extract the ARTIFACT argument, our SRE model can also correctly identify and classify the argument mention "*Coca-Cola*". We observe that the distance-based SSR learns that "*Coca-Cola*" and the verb "*buy*" in "*buy it back*" have a short syntactic distance, which may provide supporting information for the task.

For document-level EE, we show the differences in the predictions of RICB, $\text{BERT}_{\text{Large}}$, and our proposed SRE model. Table 4.9 shows two events in this paragraph. For Event 1, RICB and $\text{BERT}_{\text{Large}}$ only detect "*Colonel Ponce*" as a PERPIND argument, whereas SRE successfully detects "*Cristiani*" and "*Colonel Elena Fuente*" as well. Besides, from the original text, we can see that sentence ⟨4⟩ is very long. We observe that the SSR mechanism in SRE helps the model link "*this action*" with the following "*it*" pronouns, while the action verb "*ordered*", "*knew*", and "*carried out*" have similar estimated parsed depths. This demonstrates the usefulness of the incorporation of syntactic knowledge. Similarly, for Event 2, SRE identifies "*Mason*" in sentence ⟨7⟩ as a TARGET argument, while RICB and $\text{BERT}_{\text{Large}}$ fail. Moreover, SRE predicts "*president of the Mortgage Bank*" and a TARGET argument. Although it is not a gold argument with the corpus, it is appositive to "*Mason*", and thus also a correct argument.

4.6 Summary

In this chapter, we have introduced a Soft Syntactic Reinforcement mechanism to inject syntactic knowledge into neural EE models. To the best of our knowledge, this is the first work exploring mechanisms that enhance syntactic knowledge intrinsically captured by PLMs. Experiments on both sentence-level and document-level EE benchmark datasets have shown that our proposed SRE model outperforms state-of-the-art models on F1 for both sentence-level EE and document-level EE, and also significantly improves recall for the latter.

Chapter 5

Dynamic Task Balancing for Joint Information Extraction

While the previous chapters explore semantic and syntactic approaches to event extraction (EE) within single-task or pipeline learning frameworks, this chapter expands the focus to joint learning across multiple information extraction tasks, including EE. We introduce Adaptive Weighting Method for Joint Information Extraction (AWIE), a novel gradient-based optimization method to dynamically balance task losses for joint information extraction. We empirically show that compared to static weighting methods and existing dynamic weighting methods, it performs well and is most cost-effective. This chapter first discusses the motivation, then identifies the limitations of static weighting methods, and subsequently introduces the proposed model, followed by a discussion on the experiments and performance results.

5.1 Background

Joint Information Extraction (IE) refers to the task of extracting multiple types of informative structures (e.g., entities, relations, events, etc.) simultaneously from natural language text. It provides a unified solution for capturing the information of the text in different forms and directly benefits knowledge base construction. Figure 5.1 shows an example of joint information extraction on an input sentence. The lower layers are shared across the tasks. Joint IE therefore not only alleviates

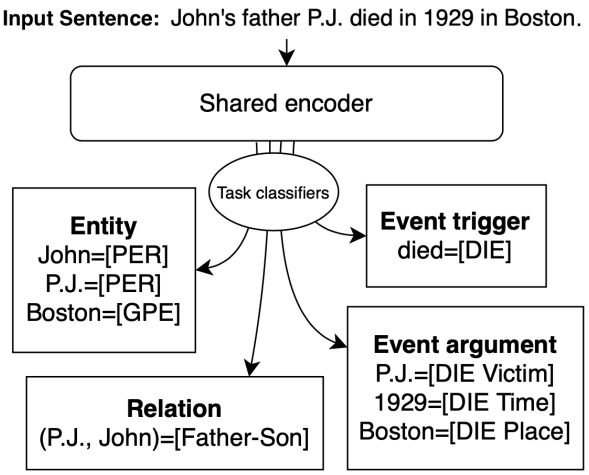

FIGURE 5.1: An example of joint information extraction for entity extraction, relation extraction, and event extraction.

the error propagation problem that is inherent to traditional pipeline IE systems, but also leads to more robust representation learning.

To date, most existing joint IE studies focus on proposing novel and highly complicated architectures. For instance, Yan et al. [122] recently proposed a HyperGraph neural network model for joint entity and relation extraction, which encodes higher-order interactions among relations and associated entities. Lin et al. [49] proposed a joint framework for four IE tasks, which innovates the decoding stage by building an optimal graph beam decoder on top of local task-specific classifiers. However, the optimization aspect of joint IE models remains under-examined. Most of the existing models are trained by optimizing a weighted linear combination of individual task losses, and the per-task weights are pre-defined and fixed throughout the training. They then rely on either random search or manual tuning to find weight combinations that lead to good performance. In the terminology of multi-task learning (MTL), this is known as *static* weighting methods, which have been observed to pose problems that hurt the performance of the learned models [74]. For example, different tasks often need different learning schedules and they may conflict with each other during the training. In this case, static weighting methods may prevent some of the tasks from being properly learned during different phases of training.

In contrast, dynamic weighting methods, in which task weights are automatically

adjusted via an algorithm, have gained much research attention in machine learning. By allowing the task losses to be balanced adaptively during the training process, such dynamic weighting methods enable the model to reach the non-convex areas of the solution space. One of the most popular dynamic weighting methods is Uncertainty [67], which uses heuristics based on homoscedastic uncertainty to balance the tasks. Moreover, Multi-Objective Optimization [74] and ParetoMTL [75] have proposed to adjust task loss weights based on task-specific gradients and search for solutions on the Pareto front. However, to the best of our knowledge, there are no previous studies on the feasibility of these weighting methods for joint IE. Most of these dynamic methods are tested mainly on synthetic toy datasets or other domains, but not on real-life NLP datasets.

In this chapter, we investigate joint IE from the optimization perspective and share the lessons learned from addressing the problems in existing works. In particular, we first identify the limitations of current static weighting methods for joint IE. Then we evaluate several dynamic weighting MTL methods for joint IE. We find that in comparison to static weighting methods, the dynamic weighting methods can find good weights cost-effectively. Furthermore, we propose a hybrid dynamic MTL method called Adaptive Weighting Method for Joint Information Extraction (AWIE), which has improvements to better fit the Joint IE problem. In particular, it accommodates user preference of a main task, without heavy computational overheads. To evaluate the proposed AWIE method, the experiments, and performance results are discussed in this chapter.

Overall, the contributions of this work are summarised as follows: (1) We empirically analyze and identify the limitations of the static weighting paradigm for Joint IE. (2) We evaluate existing dynamic weighting methods on three joint IE benchmark datasets and discuss the successes and challenges in directly applying these methods to joint IE. (3) To further enhance the performance of joint IE, we introduce a hybrid dynamic weighting algorithm, AWIE, which automatically fine-tunes the multiple tasks' weights along the learning steps. Then, we show that our proposed algorithm improves the performance over the other weighting methods on all three benchmark datasets and further allows the setting of task preference. In addition, AWIE achieves the best performance cost-effectively among all compared methods, with a low time cost and minor increase in memory cost.

5.2 Problem Formulation

We formulate Joint IE as a multi-objective optimization problem. The general form of a multi-task IE problem with m tasks is:

$$\min_{\theta} \mathcal{L}(\theta) = (\mathcal{L}_1(\theta), \mathcal{L}_2(\theta), ..., \mathcal{L}_m(\theta))^{\mathsf{T}} \tag{5.1}$$

where the trainable parameters in a model are denoted as θ and each individual IE task loss as $\mathcal{L}_i, i = 1 \dots m$. For instance, the joint extraction of entities and relations is a two-objective optimization problem:

$$\min_{\theta} \mathcal{L}(\theta) = (\mathcal{L}_{\text{NER}}(\theta), \mathcal{L}_{\text{RE}}(\theta))^{\mathsf{T}} \tag{5.2}$$

with $m = 2$, and the joint extraction of entities, relations, and events is a four-objective optimization problem:

$$\min_{\theta} \mathcal{L}(\theta) = (\mathcal{L}_{\text{NER}}(\theta), \mathcal{L}_{\text{RE}}(\theta), \mathcal{L}_{\text{TE}}(\theta), \mathcal{L}_{\text{AC}}(\theta))^{\mathsf{T}} \tag{5.3}$$

with $m = 4$. Note that, unlike single-task IE problems where optimization is performed towards a single objective, a multi-task IE problem has multiple objectives. If the relative importance of the tasks is not given, optimization is performed by looking for solutions that are *Pareto optimal*.

Definition 1 (Pareto optimality) θ^* is a Pareto optimal point for the multi-variable multi-objective function $\mathcal{L} : \mathbb{R}^n \to \mathbb{R}^m$ if there does not exist $\hat{\theta} \in \Omega$ such that $\exists i \in \{1, \dots, T\}, \mathcal{L}_i(\hat{\theta}) < \mathcal{L}_i(\theta^*)$ and $\forall j \in \{1, \dots, T\}, \mathcal{L}_j(\hat{\theta}) \leq \mathcal{L}_j(\theta^*)$, where n and m are the numbers of the model's input and output dimensions, respectively, and $\mathcal{L}$ is the model's objective. T is the number of tasks.

In other words, Pareto optimal solutions are reached if and only if there is no further gradient update step that can be taken to improve the performance on any task without hurting another task.

5.3 Static Weighting Approach

Most existing joint IE models adopt a static weighting approach, which is the most straightforward approach for MTL. It involves setting a set of fixed loss weights before training and the model is optimized for a surrogate objective that combines the individual task losses with these weights. The weights can be finetuned as hyperparameters that generate optimal model performances, but the process is expensive. Instead, many previous studies just report the use of uniform weights for tasks, such as setting all of them to 1.0 (e.g., [122–124]), or scaling others against a main task (e.g., [48, 125]). The total loss function of this method for a m-task joint IE problem is defined as follows:

$$\mathcal{L}(\theta) = \sum_{i=1}^{m} w_i \mathcal{L}_i(\theta) \tag{5.4}$$

where $\{w_i, i \in \{1, \ldots, m\}\}$ is a set of static weights with each representing the i-th task loss $\mathcal{L}_i$.

As this will restrict the relationship between the task losses to be linear, the model can only search for convex solution points. Besides, we also identify and summarize other issues that static weighting methods have as follows.

Discrepancy in convergence time When evaluating the state-of-the-art models on benchmark joint IE datasets, we notice a discrepancy between the convergence times of the different tasks. Different data types and numbers of samples lead to inconsistent learning paces (i.e., one of the tasks may start overfitting, while the other still needs further training). As shown in Figure 5.2, we conduct experiments of a baseline model (see Section 5.5.1 for details) on the SciERC dataset, which involves joint NER and RE extraction. We can see that the model's NER performance converges on the development set relatively early at around the 10th epoch (the convergence phase is represented by the green shaded area), but the learning continues to take place as RE does not reach convergence until around the 40th epoch (the convergence phase is represented by the blue-shaded area). As a result, the learning stops much later than desirable for NER, possibly overfitting the task.

Discrepancy in training loss and generalization loss Static weighting methods may assign a larger weight to task losses that are greater in magnitude, thereby

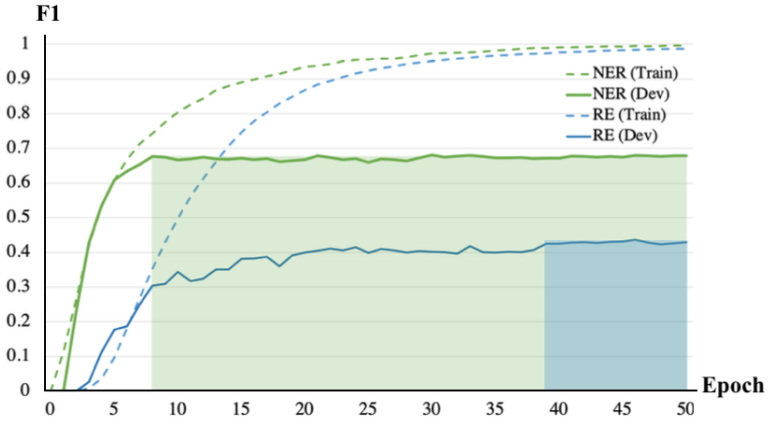

FIGURE 5.2: In multi-task IE, the learning of the tasks may require different scheduling. In this joint NER-RE model, the two tasks mature at different times.

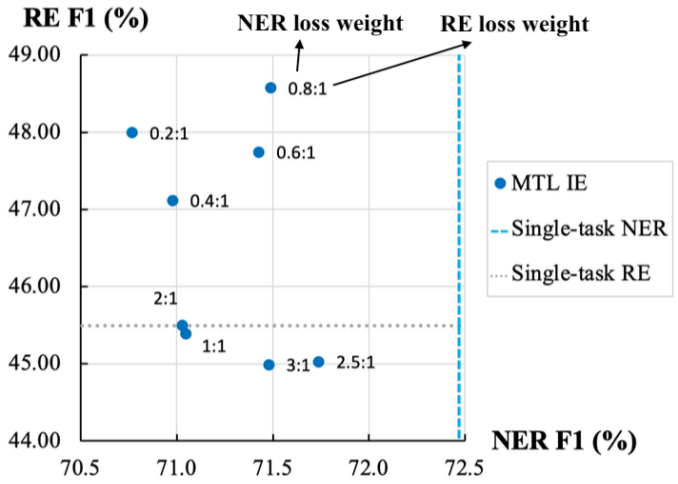

FIGURE 5.3: Performance comparison of joint NER-RE models with different fixed loss weight ratios for individual tasks (single-task baselines are shown in dotted lines). As the static weight vs task performance does not show any pattern, it is hard to optimize in the way that it can benefit mainly a "main" task by adjusting the loss weight combination.

achieving a lower total loss on the training set. However, they ignore the gap between the training loss and generalization loss. As discussed in [126], in multi-task NLP problems, it is common for training losses and generalization losses to have discrepancies in magnitudes. Besides, different tasks have different training-generalization loss patterns, such as a task with a large training loss may have the lowest generalization loss among the tasks. Due to this gap, fixing the loss weights may hurt the generalizability of a model, reducing the effectiveness of the weighting method.

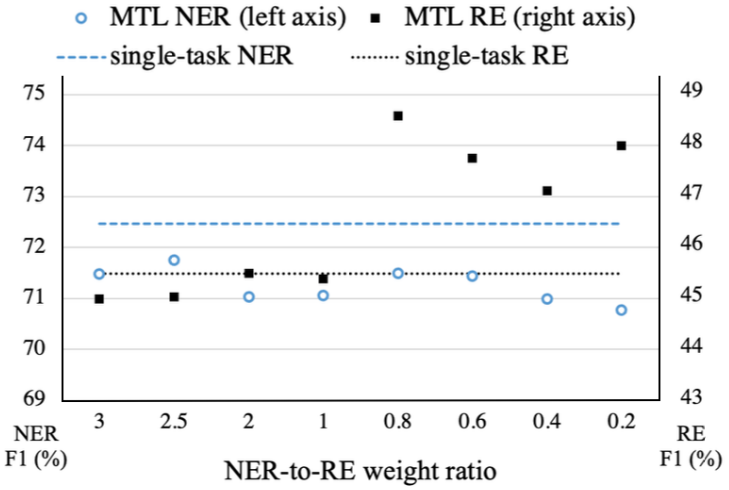

FIGURE 5.4: Model performance is often sensitive to the choice of loss weights. However, the performance results do not indicate the presence of a set of fixed loss weights that is optimal for all the tasks. Single-task NER and RE performances (dotted lines) are shown for reference.

No control over reaching the best performance for main tasks Multi-task practitioners often have a preference over tasks, i.e., there is usually a main task that is the main training target whereas other tasks are auxiliary tasks that are supposed to assist learning. The performance on the main task is by definition more important than the model performance on all the other tasks. Since there are possibilities of task conflicts, there is usually a trade-off in the optimization of all the losses. However, most existing weighting methods try to balance different tasks in the training with a set of selected weights without a systematic way to incorporate trade-off preferences.

For example, we alter the weights of NER and RE task losses for the training of a baseline model (see Section 5.5.1 for details) on SciERC. Figure 5.3 shows the scatter plot of the runs with different loss weight combinations (each point is labeled with the ratio of the actual weight for NER loss and ER loss). Figure 5.4 shows the horizontal baseline plot of the runs with different fixed loss weight ratios for individual tasks (i.e., the weight of NER loss divided by that of RE loss). We can see that it is difficult to know how to choose a specific set of weights to prioritize the tasks, as the model's performance on both tasks does not have a clear correlation pattern with the weight ratio or the weights per se.

5.4 Proposed Method

To address the issues with existing dynamic weighting methods for the joint IE problem as discussed in Section 2.3.2, we propose AWIE, a hybrid adaptive weighting method for joint IE. Following MGDA, we propose to perform a gradient descent update at time t:

$$\theta_{t+1} \leftarrow \theta_t + \eta d_t \tag{5.5}$$

where η is the overall learning rate and our goal is to find the steepest multi-objective descent direction d_t. We mainly discuss the improvements made to address the problems of existing dynamic methods.

Problem Decomposition To find d_t we first decompose the multi-task IE problem into K subproblems (as shown in Figure 2.1), following ParetoMTL [75]. Specifically, we divide the problem space with a set of preference vectors $\{\boldsymbol{u}_k | \boldsymbol{u}_k \in \mathbb{R}_+^m, k = 1, \ldots, K\}$, with each representing a different task trade-off. For a sub-region Ω_k as specified by $\boldsymbol{u}_k$, the optimisation is subject to a trade-off constraint $\mathcal{P}^{(k)}$ as follows:

$$\min_{\theta} \mathcal{L}(\theta) = (\mathcal{L}_1(\theta), \mathcal{L}_2(\theta), ..., \mathcal{L}_m(\theta))^\mathsf{T} \tag{5.6}$$

such that:

$$\begin{aligned} \mathcal{P}_j^{(k)}(\theta_t) =& (\boldsymbol{u}_j - \boldsymbol{u}_k)^\mathsf{T} \mathcal{L}(\theta_t) \leq 0, \\ \forall j =& 1, \ldots, K \end{aligned} \tag{5.7}$$

This ensures that the final solution θ_k^* found with such a constraint forms the smallest acute angle to $\boldsymbol{u}_k$ than any other vectors, i.e., the solution satisfies the specified trade-off preference. Without prior knowledge of the multi-objective problem space and to be scalable, we construct the set of preference vectors using two types of vectors: (i), the unit vector on the k-th axis as preference vectors for each $\boldsymbol{u}_k$; and (ii), an all-ones vector which represents that all the tasks are equally important. For instance, when the number of tasks $m = 2$, we take $\{\boldsymbol{u}_1 = (1, 0), \boldsymbol{u}_2 = (1, 1), \boldsymbol{u}_3 = (0, 1)\}$. If task $i \in \{1, \ldots, m\}$ is the main task, and all other tasks are auxiliary tasks, the preference vector can be set as one-hot vector

$\{\boldsymbol{u} = (0, 0, \ldots, 1, \ldots, 0, 0)$ where 1 appears at the i-th position. For each preference vector, we compute and save gradient updates separately. Although in terms of time complexity, this requires k runs for k preference vectors, we will show in the section on Experiments that empirically running AWIE once can achieve better results than the best results found via extensive random search using static weighting methods.

Algorithm 5.1: AWIE Algorithm

Input: A set of preference vectors $\boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_K$
for $k = 1, \ldots, K$ **do**
  randomly generate parameters θ_0^k
  find the initial parameters θ_1^k from θ_0^k via gradient descent
  while $t = 1, \ldots, \tau$ **do**
    compute gradients $\nabla\mathcal{L}(\theta_t)$ at time t
    solve Equation (5.8) for $\lambda_i \geq 0$ and $\beta_j \geq 0$ using Equation (5.14) and Equation (5.15)
    compute gradient update $d_t = \Delta\theta_t^{(k)} = -(\sum_i \lambda_{ti}^{(k)} \nabla\mathcal{L}_i(\theta_t) + \sum_{j \in I^{(k)}} \beta_{tj}^{(k)} \nabla\mathcal{P}_j(\theta_t))$
    perform gradient update $\theta_{t+1}^{(k)} \leftarrow \theta_t^{(k)} + \eta d_t$
  end while
end for
return Solutions with different trade-offs $\{\theta^{*(k)} | k = 1, \ldots, K\}$

Scalability Optimization Next, solving for d_t in Equation (5.6) under k preference constraints is equivalent to solving [74]:

$$(d_t^{(k)}, \alpha_t^{(k)}) = \arg \min_{d \in R^n, \alpha \in R} \alpha + \frac{1}{2}\|d\|^2 \tag{5.8}$$

such that:

$$\begin{aligned} &\nabla\mathcal{L}_i(\theta_t)^\mathsf{T} d \leq \alpha, i = 1, \ldots, m, \\ &\nabla\mathcal{P}_j(\theta_t)^\mathsf{T} d \leq \alpha, j \in \mathcal{A}_t^{(k)} \end{aligned} \tag{5.9}$$

where $\mathcal{A}_t^{(k)}$ is the set of all activated constraints for the k-th preference at input $\mathbf{x}_t$:

$$\mathcal{A}_t^{(k)} = \{j | \mathcal{P}_j^{(k)}(\theta_t) > 0\}, k = 1, \ldots, K \tag{5.10}$$

The dimension of Equation (5.8) is the number of parameters θ in the model, which makes it difficult to solve directly. Following [75, 77], we solve the Lagrangian dual problem [127] of the equation based on the Karush–Kuhn–Tucker (KKT) conditions as follows:

$$\max_{\lambda_i \geq 0} -\frac{1}{2} \left\| \sum_{i=1}^{m} \lambda_i \nabla \mathcal{L}_i(\theta_t) + \sum_{j \in \mathcal{A}^{(k)}} \beta_i \nabla \mathcal{P}_j \right\|^2 \tag{5.11}$$

such that:

$$\sum_{i=1}^{m} \lambda_i + \sum_{j \in \mathcal{A}_t^{(k)}} \beta_j = 1, \lambda_i \geq 0, \beta_j \geq 0 \tag{5.12}$$

where λ_i and β_j are the Lagrange multipliers for the linear inequality constraints in Equation (5.8). The dimension of the dual problem $|m| + |\mathcal{A}_t^{(k)}|$ only relies on the number of tasks and active constraints, which is considerably smaller than the model parameter size. For instance, in a 3-task setting where 4 trade-off constraints are activated, the dimension is $3 + 4 = 7$, regardless of the number of trainable parameters.

Equation (5.11), which is a minimum-norm problem, can be solved iteratively. It involves the computation of $f_{p,q} = (J_p^\mathsf{T} J_p, J_p^\mathsf{T} J_q, J_q^\mathsf{T} J_q)$ for each pair (J_p, J_q) as in:

$$J = (\nabla \mathcal{L}_1(\theta_t), \ldots, \nabla \mathcal{L}_m(\theta_t), \nabla \mathcal{P}_{j_1}(\theta_t), \ldots, \nabla \mathcal{P}_{j_{|\mathcal{A}|}}(\theta_t)) \tag{5.13}$$

However, ParetoMTL requires the computation of $\{f_{p,q}\}$ directly, which is a prohibitive cost for joint IE models. Therefore, we propose a converted step as follows. Notice that $\mathcal{P}_j^{(k)}(\theta_t)$ is related to $\mathcal{L}(\theta_t)$ in Equation (5.6), we first compute:

$$\begin{aligned} M_{\mathcal{L}} = (\nabla \mathcal{L}_1(\theta_t), \ldots, &\nabla \mathcal{L}_m(\theta_t))^\mathsf{T} (\nabla \mathcal{L}_1(\theta_t), \ldots, \nabla \mathcal{L}_m(\theta_t)) \\ U^{(k)} = &[(\boldsymbol{u}_1 - \boldsymbol{u}_k), \ldots, (\boldsymbol{u}_{|\mathcal{A}|} - \boldsymbol{u}_k)]^\mathsf{T} \end{aligned} \tag{5.14}$$

such that:

$$\begin{aligned} \nabla \mathcal{L}_p(\theta_t)^\mathsf{T} \nabla \mathcal{L}_q(\theta_t) &= M_{\mathcal{L}}[p, q] \\ \nabla \mathcal{L}_p(\theta_t)^\mathsf{T} \nabla \mathcal{P}_q(\theta_t) &= \langle I_k(p), M_{\mathcal{L}}[:, q] \rangle \\ \nabla \mathcal{P}_p(\theta_t)^\mathsf{T} \nabla \mathcal{P}_q(\theta_t) &= \langle U_p^{(k)} \odot U_q^{(k)}, M_{\mathcal{L}} \rangle \end{aligned} \tag{5.15}$$

TABLE 5.1: Statistics and data split of the datasets.

Dataset	SciERC	ACE05-R	ACE05-E
Domain	AI	News	News
#Train Documentefs	350	351	529
#Dev Documents	50	80	30
#Test Documents	100	80	40
#Total Documents	500	511	599
#End Tasks	2	2	3
#Entity Types	6	7	7
#Relation Types	7	6	-
#Trigger Types	-	-	33
#Argument Types	-	-	22

where $I_k(p)$ is the p-th row of a $k \times k$ Identity matrix, $\odot$ denotes an outer product and $\langle \cdot, \cdot \rangle$ denotes an inner product. [1] The overall Algorithm of AWIE is shown in Figure 5.1.

5.5 Experiments

In this section, we describe our experimental setup and evaluation results of the weighting methods on three benchmark joint IE datasets.

5.5.1 Experimental Setup

Datasets We evaluate our proposed method on three popular multi-task IE benchmark datasets: SciERC, ACE05-R, and ACE05-E. The SciERC dataset[2] consists of scientific abstracts annotated with scientific entities and relations. ACE05-R and ACE05-E are two datasets based on the Automatic Content Extraction (ACE) 2005 English corpus[3], which contains news documents from mixed-genre sources. The ACE05-R dataset shares the same task setting as SciERC, i.e., two-task joint extraction of entity and relation. The ACE05-E dataset requires joint

[1] This reduces up to 98% running steps and 85% of computational memory for each Pareto update in practice.

[2] `http://nlp.cs.washington.edu/sciIE/`

[3] `https://catalog.ldc.upenn.edu/LDC2006T06`

learning of all four tasks in IE, which includes entity, relation, and event annotations. For all three datasets, we follow the preprocessing in [48]. Table 5.1 shows the statistics and data split of the datasets.

Evaluation Metrics Following existing works, we use micro-average F1 as the evaluation metrics. Entities, triggers, and arguments are treated as correct if and only if both the type label and the boundary of a predicted mention match with a gold one. Relations are treated as correct if and only if the boundaries of both entities and the relation type match a gold one.

Model Architecture We conduct experiments on a classical joint IE model, which is a simplified version of Wadden et al.'s architecture [48]. The main differences between our model and DYGIE++ are: (i) we skip the graph propagation mechanism; and (ii) we do not use the coreference resolution task for auxiliary learning. Figure 5.5 illustrates the architecture of our model, which consists of three stages:

- **BERT Encoding** - The input is tokenized and encoded using BERT token representation with a sliding window approach. Each document is split into a list of sentences of size L and subsequently fed to BERT: $\mathbf{h}_{1..L} = BERT(\mathbf{x}_{1..L})$
- **Span Enumeration** - The model enumerates text spans that are candidates for classification. Each span is represented by the tokens at the left and right endpoints of the span (e.g., if a span of text starts with the a-th token and ends at the b-th token), followed by a learnable span width representation: $\mathbf{g}_{(a,b)} = [\,\mathbf{h}_a, \mathbf{h}_b, f(a,b)]$
- **Span Classification** - The task-specific prediction component for each end task consists of a two-layer feedforward neural network (FFNN) that is used to score span or span pair representations. For trigger and named entity prediction for span $\mathbf{g}_i$, we compute $FFNN_{task}(\mathbf{g}_i)$. For relation and argument role prediction, we concatenate the relevant pair of embeddings and compute $FFNN_{task}([\mathbf{g}_i\ ,\ \mathbf{g}_j])$.

In summary, the model embeds all the tokens using a trainable BERT-based encoder and generates span representations by taking the first and last token representations along with a trainable span length embedding. Then, it uses two

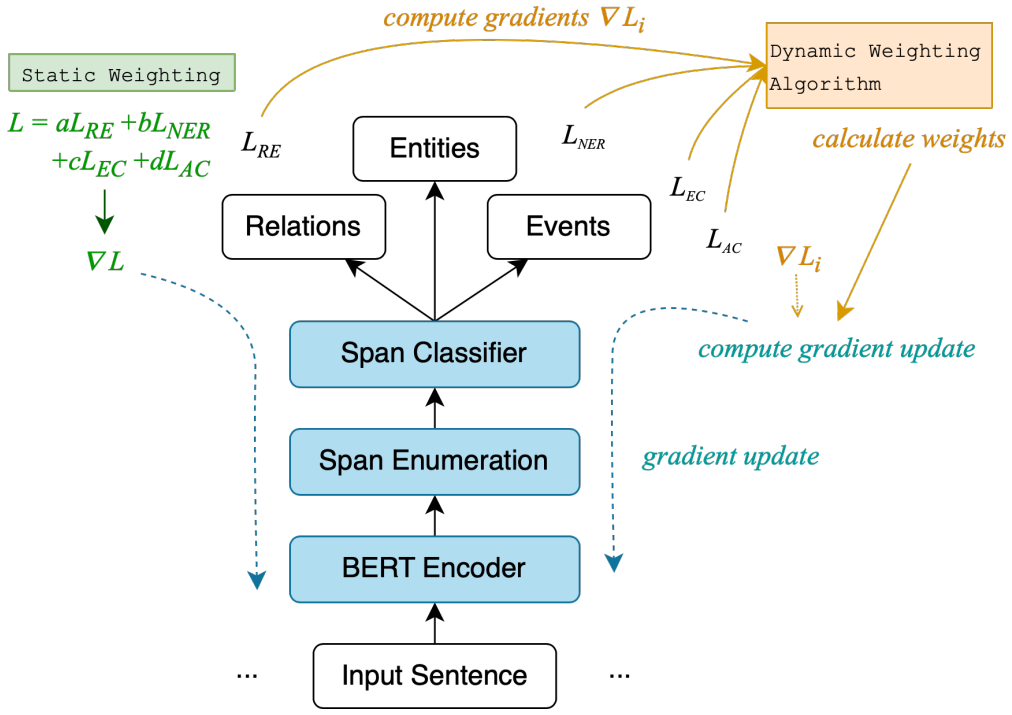

FIGURE 5.5: Model architecture for the joint IE experiments.

consecutive feed-forward network layers as a task-specific classifier for each task. Therefore, the shared layers are BERT Encoding and Span Enumeration. The individual task loss function $L = CrossEntropyLoss(\mathbf{y}_{true}, \mathbf{y}_{pred})$. The overall loss function is a weighted-sum combination of all the individual task losses, as formulated by Equation (5.4).

Implementation Details We implement our proposed method based on AllenNLP and PyTorch. For SciERC, the encoder is SciBERT cased [82]. For ACE05-R and ACE05-E, the encoder is RoBERTa base. We follow the preprocessing code in the original DYGIE++ repository [48]. We set the maximum epoch to 50 and 5 epochs for early stopping patience. For the initial solution search, we set the threshold ϵ to be 1e-20 and the maximum number of iterations to 500. We train all models on a single GPU. The learning rate is set to be 5e-5 for the encoder and 1e-3 for other trainable parameters. We adopt the slanted triangular learning rate schedule for the optimizer.

Compared Methods For static weighting optimization methods, we first include the results reported in the original DYGIE++ paper as one of the baseline

TABLE 5.2: Performance results (%) of weighting methods on SciERC. NER denotes the entity extraction task, and RE denotes the relation extraction task.

	Task weights		NER	RE
	NER	RE	F1	F1
DYGIE++ (manual tuning)	0.2	1.0	67.2	46.7
DYGIE++ (replicated)	0.2	1.0	67.0±0.70	47.5±0.91
Single task	1	-	67.4±0.80	-
	-	1	-	45.7±1.17
Uncertainty			68.1±0.51	46.8±0.44
MGDA-UB	(*dynamic*)		68.0±0.61	43.2±0.83
AWIE (ours)			**68.9±0.48**	**48.8±0.98**

methods and refer to it as "DYGIE++ (manual tuning)". In addition, we replicate the results of the baseline model with the optimal weights reported by [48], which we refer to as "DYGIE++ (replicated)", as well as on single task data, which we refer to as "Single task".

For dynamic weighting optimization methods, we use Uncertainty and MGDA-UB. ParetoMTL is excluded because the computational cost of the method on the baseline model is prohibitive. Note that the dynamic weighting methods perform adaptive gradient updates at each back-propagation step and the task loss weights are updated. Therefore, there are no fixed task weights pre-defined for the training stage.

5.5.2 Experimental Results

Performance Results based on SciERC Table 5.2 shows the performance results based on SciERC. From the results, we can see that on SciERC, Uncertainty and MGDA-UB achieve 68.1% and 68.0% in F1 respectively for NER, which are both better than the best static weighting method result (67.4%). Both methods achieve lower results for RE than the replicated DYGIE++ baseline (47.5% in F1), but Uncertainty still performs slightly better than the original DIGYE++ and the RE single-task baseline. On SciERC, our proposed AWIE method achieves 68.9% and 48.8% in F1 for NER and RE, respectively, which are higher on both tasks than the replicated DYGIE++ (1.7% for NER and 2.1% for RE) and the single-task baselines (1.5% for NER and 1.3% for RE). On both tasks, it also outperforms DYGIE++ with the manual tuning method and the other dynamic methods.

TABLE 5.3: Performance results (%) of weighting methods on ACE05-R. NER denotes the entity extraction task, and RE denotes the relation extraction task.

	Task weights		NER	RE
	NER	RE	F1	F1
DYGIE++ (manual tuning)	0.2	1.0	86.3	64.8
DYGIE++ (replicated)	0.2	1.0	87.0±0.34	64.9±0.79
Single task	1	-	87.4±0.17	-
	-	1	-	62.7±0.94
Uncertainty			87.2±0.31	65.1±0.49
MGDA-UB	(*dynamic*)		87.4±0.41	63.3±1.11
AWIE (ours)			**88.3±0.39**	**68.1±0.51**

TABLE 5.4: Performance results (%) of weighting methods on ACE05-E. NER denotes the entity extraction task, and RE denotes the relation extraction task. For event extraction, TE denotes the event trigger extraction task and AC denotes the argument classification task.

	Task weights				NER	RE	TE	AC
	NER	RE	TE	AC	F1	F1	F1	F1
DYGIE++ (manual tuning)	0.5	0.5	0.2	1.0	89.3	55.1	70.0	50.0
DYGIE++ (replicated)	0.5	0.5	0.2	1.0	89.0±0.46	52.9±3.49	67.8±1.97	50.5±2.20
Single task	1	-	-	-	88.9±0.50	-	-	-
Single task	-	1	-	-	-	53.3±1.17	-	-
Single task	-	-	1	-	-	-	66.4±0.32	-
Single task	-	-	-	1	-	-	-	7.04±3.39
Uncertainty					89.4±0.67	55.0±1.95	69.8±1.57	51.1±1.09
MGDA-UB	(*dynamic*)				88.1±1.18	38.8±3.63	67.1±3.09	43.8±5.70
AWIE (ours)					**90.1±0.46**	**57.3±1.26**	**71.7±1.33**	**53.1±1.07**

Performance Results based on ACE05-R Table 5.3 shows the performance results based on ACE05-R. From the results, we can see that on ACE05-R, the patterns are similar to the results based on SciERC. More specifically, Uncertainty achieves 87.2% in F1 for NER and 65.1% in F1 for RE, while MGDA-UB achieves 87.4% and 63.3% in F1 on the two tasks respectively. Both achieve the same or better results for NER compared to the static weighting methods. While MGDA-UB achieves higher F1 for NER, Uncertainty achieves the second-best F1 in RE among all the methods except AWIE. For performance comparison on the ACE05-R dataset, AWIE achieves 88.3% in F1 for NER and 68.1% for RE, both higher than the best result obtained for NER via manual tuning and comparable to that for RE. In comparison with the single-task baselines, AWIE improves the performance by 1.3% and 3.2% in F1, respectively. It also significantly outperforms Uncertainty and MGDA-UB on both tasks.

TABLE 5.5: Comparison of empirical computational time and memory costs of the methods on SciERC.

Methods	Training Cost	
	Time (hour)	Memory (Gigabytes)
Random Search 50	15.1	2.7
Uncertainty	1.5	2.7
MGDA-UB	2.0	3.1
AWIE	4.9	2.8

Performance Results based on ACE05-E Table 5.4 shows the performance results based on ACE05-E. On ACE05-E, we observe that Uncertainty still achieves strong NER performance with an F1 of 89.4%. It also achieves a high F1 of 51.1% on the AC task. Its performance in RE and TE is reasonably good compared to static weighting methods. While MGDA-UB performs worse than the replicated DYGIE++ on RE and AC, it achieves fair results on NER (88.1% in F1) and TE (67.1% in F1). In comparison, on the ACE05-E dataset, AWIE achieves 90.1% in F1 for NER, improving DYGIE++ by 0.8%. For Event Extraction, it achieves 71.7% in F1 for EC and 53.1% for AC, which is higher than the DYGIE++ baseline by 1.7%-3.1%. We observe that AWIE is significantly better than the single-task baselines on all tasks. Moreover, AWIE's performance on the tasks is higher or comparable to the performance of the dynamic weighting baselines. In particular, it outperforms Uncertainty on AC by 2.0% in F1.

Overall Results Overall, we observe that dynamic weighting methods can improve the performance on different tasks across the three datasets. In addition, compared to static weighting methods and single-task baselines, which require extensive hyperparameter tuning often taking more than 30 runs, dynamic weighting methods can achieve reasonably good results in a single run. This demonstrates their advantages over the static weighting methods.

We can also see that our proposed AWIE method consistently improves over the corresponding baseline methods on all tasks across different datasets, with absolute F1 improvement varying from 0.7%-4.0%. Compared to the single-task models, multi-task learning is beneficial to all tasks, improving the model by 0.6%-4.3% in F1. Although random search can generate solutions with high-performance results, our method can find comparable or better solutions in all task settings with much fewer trials.

5.5.3 Computational Cost Analysis

In Table 5.5, we compare the computational costs of our proposed AWIE method with the static method (Random Search of 50 trials) and two state-of-the-art dynamic weighting methods, Uncertainty and MGDA-UB, on SciERC. Table 5.5 shows that our proposed AWIE method, along with the other two dynamic weighting methods, is more time-efficient than the Random Search 50 method in terms of time cost. The training time of AWIE is around 4.9 hours on SciERC, which is 67.5% faster than the Random Search 50 method. Additionally, we also observe that the other dynamic weighting methods require more computational space to compute the dynamic weights, but such a cost is low. Specifically, AWIE incurs a slight increase in memory requirement of around 4% compared to the static method. The results suggest that while the dynamic weighting methods typically incur higher costs due to the need to store and compute intermediate results and gradients across multiple tasks, they benefit from more efficient learning, leading to faster and better convergence. In conclusion, our proposed AWIE, which achieves the best performance among all the compared methods, is much faster both during training and inferencing with comparable requirements of memory.

5.6 Summary

In this chapter, we have explored the feasibility of dynamic weighting MTL methods for joint IE, besides identifying the limitations of the static weighting approach, which is used in existing joint IE works. We have proposed AWIE, a hybrid dynamic weighting method for joint IE, which automatically balances the tasks by assigning weights for their losses based on multi-objective gradient descent. We have conducted experiments to compare the performance of the static weighting methods, state-of-the-art dynamic weighting methods, and our proposed AWIE method. The experimental results on three datasets have shown that dynamic methods are useful for achieving good results within a single run. Moreover, we have shown that our proposed AWIE method can dynamically assign effective weights to the losses of individual tasks and outperform the baselines. Overall, we have shown that our method achieves good performance and is cost-efficient, with the ability

to accommodate user task preferences by generating solutions with different task trade-offs.

Chapter 6

Biomedical Event Trigger Extraction

To study the effectiveness and generalizability of the proposed mechanisms presented in previous chapters, we applied them to the biomedical domain. The biomedical domain-specific event model, Bio-SemSyntEE, combines the semantic-based and syntax-based mechanisms for biomedical event trigger extraction. In this chapter, we discuss the background and then present the proposed model, experimental setup, and performance results.

6.1 Background

In this chapter, we apply the mechanisms from the proposed models discussed in previous chapters to a domain-specific application. The primary objective is to evaluate the performance of these mechanisms in extracting events from text within the specific domain and to compare their efficacy against both discriminative and generative state-of-the-art models. More specifically, we investigate event trigger extraction in the biomedical domain and present a model named Bio-SemSyntEE, that combines our proposed mechanisms for SemPRE and SRE as discussed in Chapter 3 and Chapter 4, respectively, on biomedical EE benchmark datasets. This investigation provides insights into the strengths and limitations of our proposed mechanisms of our models in this practical application and highlights areas for future improvement.

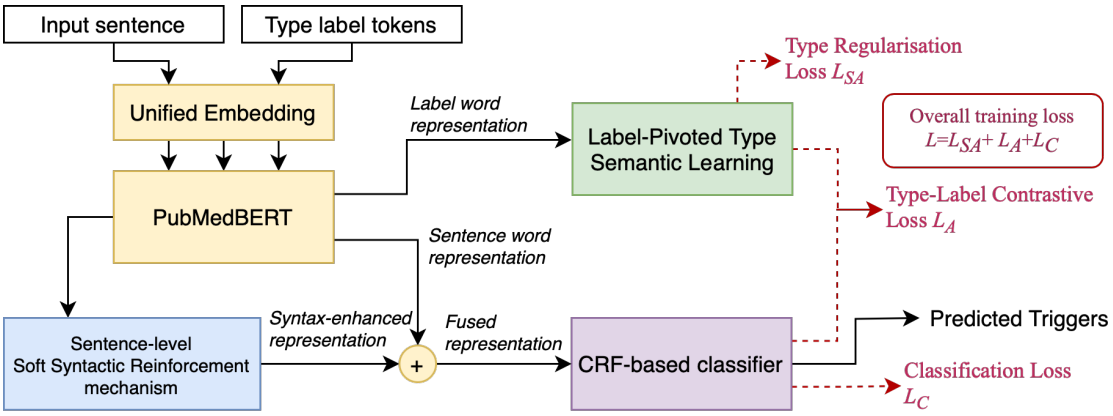

FIGURE 6.1: Bio-SemSyntEE model architecture.

Overall, the main contributions of the work in this chapter are summarized as follows: (1) We apply previously proposed SemPRE and SSR mechanisms to biomedical event trigger extraction by training a domain-specific EE model, Bio-SemSyntEE, that combines our proposed mechanisms for SemPRE and SRE with PubmedBERT. (2) We evaluate our model on three benchmark datasets for Biomedical EE and achieve state-of-the-art performance. We show that our Bio-SemSyntEE model achieves state-of-the-art performance, and each of our proposed mechanisms contributes to performance improvement.

6.2 Proposed Model

For the domain-specific Bio-SemSyntEE model, we combine the following techniques proposed in Chapters 3 and 4:

- **SemPRE** - It consists of a unified input-label encoding mechanism, a contrastive type semantic pivoting mechanism, and a CRF-based decoder for trigger classification. Same as the general domain SemPRE model introduced in Chapter 3, our domain-specific model takes an input sequence that concatenates the input sentence and pre-defined event type labels. The input sequence is passed through a PubMedBERT-based encoder, which performs unified input-label encoding. The encoder is connected to a contrastive learning module that learns input and label semantics via an input-label interaction matrix and a type relationship matrix. A type regularization loss

is added to prevent overfitting of the type relationship matrix. We use a Conditional Random Field as the task decoder.

- **SRE** - We use the depth-based[1] Soft Syntactic Reinforcement (SSR) module as proposed in Chapter 4, which is built upon the encoder used in SemPRE. We perform pre-training of the encoder based on PubMedBERT and the SSR module based on the Penn Tree Bank dependency dataset[2]. The syntactic reinforced representation generated by the SSR module is combined with the sentence representation generated by the unified encoder.

Figure 6.1 shows the overall architecture of our Bio-SemSyntEE model.

6.3 Experiments

In this section, we describe the datasets, evaluation metrics, implementation details, and compare models, and present the experimental results.

6.3.1 Experimental Setup

Datasets We utilize three domain-specific datasets for evaluation:

- **MLEE** - It is a biomedical event extraction dataset consisting of over 2600 abstracts from PubMed, focusing on molecular biology events. There are 29 event types, defined according to different levels of events in the Gene Ontology (GO).
- **BioNLP GENIA 2009 (GE09)** - It is the most widely used version of biomedical datasets based on the GENIA corpus, containing more than 1200 abstracts from PubMed. It consists of 9 event types, which are defined based on the GENIA ontology and focus on protein biology.

[1] As discussed in Chapter 4, we observed that depth-based SSR achieves better results than other variations on the sentence-level event trigger extraction task.

[2] In the future, we intend to explore GENIA Tree Bank [128], which is a biomedical domain syntactic dependency parse dataset.

Table 6.1: Dataset statistics of MLEE, GENIA 2009, and GENIA 2011.

Dataset	#Document	#Instance	#Event Type	#Event
MLEE	262	286	29	6575
GENIA 2009	1210	11,346	9	13,623
GENIA 2011	960	1375	9	13,537

- **BioNLP GENIA 2011 (GE11)** - It is a corpus of around 1000 abstracts and full papers from PubMed, designed for training and evaluating biomedical event extraction models. It consists of 9 event types.

Each dataset is preprocessed to ensure consistency and compatibility with the input requirements of the event trigger extraction task. The statistics of the datasets are shown in Table 6.1.

Evaluation Metrics We use precision (P), recall (R), and micro-average F1 as the evaluation metrics. Micro-average F1 is reported as opposed to macro-average F1 because it better reflects performance across all classes in the biomedical datasets with imbalanced distributions. More specifically, we use the BIO tagging scheme to mark the trigger candidate boundary. We calculate the scores when both the boundary and the event types of a predicted trigger match the gold ones.

Implementation Details We implement the code based on Pytorch and Hugging Face transformers library. We utilize Microsoft's PubMedBERT base[3] as the encoder. We fine-tune the model using a learning rate of 3e-5. The maximum sentence length is set to 256 and the batch size is set to 16. Dropout is set to 0.1 to prevent overfitting, and we use the AdamW optimizer with epsilon set to 1e-6, beta 1 set to 0.9, and beta 2 set to 0.999. The training maximum epoch is set to 50. We apply early stopping with a patience of 10 epochs.

Compared Models We compare our Bio-SemSyntEE model against the following SOTA models:

[3] `https://huggingface.co/microsoft/BiomedNLP-BiomedBERT-base-uncased-abstract`

- **Bio-SVM** [88] - It is the state-of-the-art feature-based discriminative model for Biomedical Event Extraction, which combines syntactic and semantic features with biomedical domain knowledge representation.
- **BiLSTM-FastTest** [92] - It is a BiLSTM-based model that utilizes FastText word representations for the task.
- **DeepEventMine** [94] - It is the end-to-end domain-specific model for Biomedical Event Trigger Extraction.
- **TEES-CNN** [91] - It is a CNN-based pipelined model that extends the Turku Event Extraction System (TEES), a classical text mining program.
- **DEGREE-E2E** [52] - It is the state-of-the-art generative model for end-to-end event extraction. It learns to summarize events mentioned in a paragraph and utilizes weakly-supervised information to facilitate template-based generation.

In addition, to address the rising interest in general-purpose Large Language Models (LLMs), we also include the results of ChatGPT-3.5 and ChatGPT-4 in the zero-shot and few-shot (5-shot) learning settings.

6.3.2 Experimental Results

Performance Results As shown in Table 6.2, we compare the performance of our Bio-SemSyntEE model against zero-shot and few-shot (5-shot) large language models, namely ChatGPT-3.5 and ChatGPT-4, and the fully supervised models based on the MLEE, GE09, and GE11 datasets.

The performance results show that our Bio-SemSyntEE model demonstrates superior performance across all datasets. On the MLEE dataset, it achieves the precision, recall, and F1 of 80.86%, 80.92% and 80.89%, respectively. For the GE09 dataset, our model achieves the precision, recall, and F1 of 72.15%, 73.06% and 72.61%, respectively. For the GE11 dataset, the precision, recall, and F1 are 74.90%, 71.67%, and 73.25%, respectively. These results indicate that our model not only significantly outperforms zero-shot and few-shot large language models but also surpasses existing state-of-the-art fully supervised models on the biomedical event trigger extraction task. More specifically, the zero-shot large language

TABLE 6.2: Performance results (%) for Biomedical Event Trigger Extraction based on MLEE, GE09 and GE11. The best performance for each column is highlighted in bold.

Methods	MLEE			GE09			GE11		
	P	R	F1	P	R	F1	P	R	F1
Zero-Shot Large Language Models									
ChatGPT-3.5	33.02	30.17	31.53	17.53	26.51	21.10	14.69	28.00	19.27
ChatGPT-4	35.40	34.48	34.93	17.92	27.01	21.55	15.28	29.33	20.09
Few-Shot Large Language Models									
ChatGPT-3.5	43.75	40.24	41.92	20.54	29.50	24.22	23.53	32.00	27.12
ChatGPT-4	44.63	42.10	43.33	21.46	31.07	25.39	24.51	33.33	28.25
Fully Supervised Models									
Bio-SVM	75.56	**81.29**	78.32	-	-	-	-	-	-
BiLSTM-FastText	77.89	78.28	78.08	68.21	58.55	63.01	68.44	65.26	66.81
DeepEventMine	79.37	78.86	79.12	-	-	-	72.05	68.89	70.43
TEES-CNN	81.49	78.43	79.93	-	-	-	73.32	68.72	70.95
DEGREE-E2E	-	-	70.20	61.07	56.60	58.75	-	-	59.20
Bio-SemSyntEE	**80.86**	80.92	**80.89**	**72.15**	**73.06**	**72.61**	**74.90**	**71.67**	**73.25**

models, ChatGPT-3.5 and ChatGPT-4, achieve notably low performance across all datasets. For example, zero-shot ChatGPT-4 achieves F1 of only 34.93% on the MLEE dataset, which is much lower than that of Bio-SemSyntEE. ChaptGPT-4 also achieves lower F1 than that of Bio-SemSyntEE on GE09 and GE11. Moreover, few-shot large language models also fail to achieve satisfactory results. Despite being provided a few examples for fine-tuning, ChatGPT-3.5 and ChatGPT-4 perform badly, with F1 being lower than 50% on MLEE and lower than 30% on GE09 and GE11. This further shows that the Biomedical Event Trigger Extraction task is still too challenging for the general-purpose large language models and that domain-specific knowledge and training data are crucial for models to excel in biomedical event trigger extraction tasks.

In comparison to the five fully supervised biomedical EE models, our Bio-SemSyntEE model also outperforms all of them in precision, recall, and F1 across datasets. For example, Bio-SemSyntEE demonstrates an improvement of 2.81%, 9.60%, and 6.44% in F1 over the BiLSTM-FastText model on the MLEE, GE09 and GE11 datasets, respectively. On the most widely used MLEE dataset, Bio-SemSyntEE outperforms Bio-SVM by 2.57%, DeepEventMine by 1.77%, and TEES-CNN by 0.96%, respectively, in F1. The general-domain model DEGREE-E2E reaches F1 of 70.20% on MLEE, 58.75% on GE09, and 59.20% on GE11, which are 10.69%, 13.86%, and 14.05% lower than those of Bio-SemSyntEE, respectively.

TABLE 6.3: Performance results (%) on the three datasets in the ablation studies. ΔF1 indicates the difference from the original model.

Model	MLEE		GE09		GE11	
	F1	Δ F1	F1	Δ F1	F1	Δ F1
Bio-SemSyntEE	80.89	-	72.61	-	73.25	-
w/o SRE	79.58	-1.31	71.42	-1.19	71.94	-1.32
w/o SemPRE	79.02	-1.87	70.79	-1.82	72.08	-1.17
w/o PubMedBERT	74.35	-6.54	60.74	-11.87	62.96	-10.29

In conclusion, the results show that our Bio-SemSyntEE model is the best performer among the compared models. Its high precision, recall, and F1 across the three benchmark datasets highlight its robustness and effectiveness for biomedical domain event trigger extraction.

Ablation Studies As shown in Table 6.3, we conduct ablation studies to show the impact of various components on the performance of our Bio-SemSyntEE model in biomedical domain event trigger extraction. Bio-SemSyntEE achieves 80.89%, 72.61%, and 73.25% in F1 on the MLEE, GE09, and GE11 datasets, respectively. Removing the SRE mechanism ("w/o SRE") results in a performance drop of 1.31%, 1.19%, and 1.32% in F1 on the MLEE, GE09, and GE11 datasets, respectively. This indicates that the SRE mechanism effectively enhances the model's ability to extract and classify events accurately. Removing the SemPRE mechanism ("w/o SemPRE") causes the performance to decrease by 1.87%, 1.82%, and 1.17% in F1 on the MLEE, GE09, and GE11 datasets, respectively. This suggests that the SemPRE mechanism also benefits biomedical event trigger extraction. Both our proposed SRE and SemPRE mechanisms for general-domain event extraction are generalizable to the biomedical domain.

Moreover, we also observe that replacing the domain-specific pre-trained model with a general-purpose pre-trained model such as BERT ("w/o PubMedBERT") results in a substantial performance drop. The ΔF1 are -6.54%, -11.87%, and -10.29% on the MLEE, GE09, and GE11 datasets, respectively. This significant reduction shows that domain-specific pre-training is critical to event extraction, as it captures the nuances and complexities of biomedical texts, which general models like BERT cannot adequately do. Overall, each component of the Bio-SemSyntEE model, namely SRE, SemPRE, and domain-specific pre-trained model

PubMedBERT, collectively contributes to its good performance and effectiveness for biomedical event trigger extraction.

6.4 Summary

In this chapter, we have applied the mechanisms from our proposed models to a domain-specific application and presented the Bio-SemSyntEE model for biomedical event trigger extraction. We have conducted experiments to compare its performance against discriminative and generative state-of-the-art models, including general-purpose LLMs. Experimental results on three benchmark datasets show that Bio-SemSyntEE outperforms other models, demonstrating the effectiveness and good generalisability of the mechanisms from our models for biomedical event trigger extraction. In addition, the results suggest that both semantic-based and syntax-based mechanisms are beneficial to the domain-specific application.

Chapter 7

Conclusion and Future Work

In this chapter, we summarize our works in the previous chapters and discuss the future work.

7.1 Summary

Events are important components of human cognition and communication, encapsulating rich information about actions, entities, and relationships. In the big data era, the ability to automatically process unstructured textual data and extract events of concern has become increasingly essential across diverse domains such as media, business, cybersecurity, and biomedical research. Traditional methods for event extraction often struggle with the inherent complexity and variability of natural language, driving a surge of interest in leveraging deep learning and neural network techniques.

Motivated by the growing significance of event extraction and the potential of neural approaches, this thesis investigates novel methods to enhance event extraction using deep learning techniques. By focusing on event representation learning through semantic and syntactic knowledge injection and multi-task learning optimization, the thesis aims to advance the current state-of-the-art in event extraction and facilitate more accurate and comprehensive event identification and understanding.

In summary, the research has made the following contributions:

- For semantics-based representation learning for Event Detection (ED), we have proposed SemPRE, a novel semantic learning model tailored for ED, which offers semantic clues to enhance the task by integrating label semantic information. SemPRE utilizes pre-defined event type labels to extract semantic representations of event types. By employing a unified input-label representation learning architecture, SemPRE achieves significant performance enhancements compared to existing ED models. Notably, SemPRE achieves these improvements without requiring additional annotated data or relying on external linguistic resources. We have demonstrated the robustness of SemPRE across various challenging scenarios, including instances with limited training data, multiple events within a sentence, and ambiguous trigger words.

- For syntax-based reinforcement for Event Extraction (EE), we have proposed SRE, the Soft Syntactic Reinforcement model for Neural EE. We have introduced a Soft Syntactic Reinforcement mechanism aimed at enriching syntactic knowledge within pre-trained language models for EE tasks. Experimental evaluations on both sentence-level EE and document-level EE benchmark datasets have validated the effectiveness of our proposed method, surpassing state-of-the-art models in terms of F1 and notably enhancing recall for document-level EE. This contribution represents a significant advancement in leveraging syntactic information for the neural event extraction approach.

- For dynamic task balancing for joint Information Extraction (IE), we have identified the limitations of static weighting approaches commonly used in existing joint IE works and investigated the feasibility of dynamic weighting Multi-Task Learning (MTL) methods for joint IE. Furthermore, we have proposed AWIE, a hybrid dynamic weighting method for joint IE. AWIE dynamically balances tasks by assigning weights to their losses based on multi-objective gradient descent. Experimental results on three datasets have demonstrated the effectiveness of dynamic weighting methods in achieving superior results within a single run. Particularly, we have demonstrated that AWIE outperforms existing baselines by dynamically assigning effective weights to task losses, thereby improving overall performance. Moreover, AWIE offers flexibility in accommodating user task preferences by generating solutions with different task trade-offs. Cost analysis shows that it is reasonably time-efficient and memory-efficient.

- For domain-specific evaluation of the methods on biomedical datasets, we have proposed Bio-SemSyntEE, a biomedical event trigger extraction model that incorporates our proposed semantic-based and syntax-based mechanisms. Experimental results have shown that Bio-SemSynt achieves state-of-the-art performance across three biomedical-domain benchmark datasets, outperforming existing generative and discriminative models. We have demonstrated the effectiveness and generalisability of our proposed mechanisms. Moreover, Bio-SemSyntEE utilises the domain-specific pre-trained encoder PubMedBERT and the experimental results have shown that the mechanisms from our proposed SemPRE and SRE models are compatible with the domain-specific pre-trained encoder and both mechanisms lead to significant performance improvements of the model.

7.2 Future Work

In this section, we discuss three feasible and meaningful directions for future work on deep learning event extraction, namely generative methods for EE, meta learning for joint IE, and few-shot EE.

7.2.1 Generative Methods for Event Extraction

In this thesis, we mainly focus on discriminative EE methods. With the rapid development of large language models (LLMs) such as T5 [129], GPT [130], and Llama [131], generative EE methods, which frame Event Extraction (EE) as a sequence-to-sequence problem, are gaining increasing interests. Generative methods offer a more flexible output format compared to discriminative approaches. While current state-of-the-art EE methods are predominantly discriminative, the generative approach shows promise due to its greater flexibility in output representation.

Based on this thread of studies, future research in generative methods for event extraction could explore several avenues for further improvement. Investigating novel architectures and training strategies tailored to the unique characteristics of event extraction tasks could lead to more effective and efficient generative models.

For example, PGAD [132] proposed a text diffusion model to generate context-aware prompt representations for event argument extraction. Code4Struct [133] converted text into code and utilized the capability of LLMs to handle structured prediction tasks. By incorporating programming language features, the model introduced external knowledge and constraints by aligning the structure with the generated code. Besides, exploring techniques to incorporate additional linguistic features or domain-specific knowledge into the training process could enhance the ability of generative models to capture complex event semantics.

7.2.2 Meta Learning for Joint Information Extraction

In Chapter 5, we have proposed to use an adaptive task balancing method for optimising Joint IE learning. Our proposed method which is based on first-order gradient descent achieves good performance. However, task relationships can be complex and dynamic, requiring more sophisticated approaches to effectively capture and leverage these dependencies [134, 135]. To address this challenge, future work could explore the application of meta-learning techniques to enhance the adaptability and generalization of Joint IE models across diverse tasks and domains. By incorporating meta-learning, we could potentially develop models that can quickly adjust to new tasks with minimal data, thereby improving the efficiency and robustness of the learning process.

Meta-learning, often referred to as "learning to learn," involves training models in a way that allows them to learn new tasks more rapidly and with fewer data points. In the context of Joint IE, future research could investigate how meta-learning can provide significant benefits by enabling the model to better understand and exploit the underlying structure of different tasks. This approach could help in fine-tuning the model's parameters to be more sensitive to task-specific nuances, leading to improved performance across varied IE scenarios. Moreover, by leveraging meta-learning, we could create a more flexible and scalable framework for Joint IE, capable of adapting to the evolving nature of information extraction tasks in real-world applications. This enhancement would not only boost the model's accuracy but also reduce the computational resources required for training, making it a practical solution for large-scale and dynamic environments.

7.2.3 Few-Shot Event Extraction

In Chapter 3 and Chapter 4, we have investigated semantics-based and syntax-based EE methods, which are fully supervised learning. Few-shot event extraction, in contrast, represents a promising direction for future work, which reduces the dependency on large annotated datasets. In few-shot event extraction, the goal is to develop models that can accurately identify and extract events from text with only a small number of annotated examples per event type. This is of importance to many domains as generating a large-scale annotated dataset for a specific problem is often expensive. Ma et al. [136] evaluated 12 existing methods on three datasets for few-shot Event Detection, and suggested the potential of leveraging LLMs for few-shot ED.

Few-shot learning leverages techniques such as meta-learning, transfer learning, and prototype networks to generalize from limited examples, and future work in Event Extraction (EE) could adapt these techniques to train models that handle new event types with minimal annotated data by capturing shared structures and semantics. Integrating few-shot learning into EE promises practical and scalable solutions, especially in data-scarce domains, and future research should develop robust evaluation frameworks and explore combining few-shot learning with existing methods to create accurate, efficient, and versatile models for diverse real-world event data.

List of Author's Awards, Publications and Submitted Works

Awards

- **Best Student Paper Award**, "Semantic Pivoting Model for Effective Event Detection." at *the 14th Asian Conference on Intelligent Information and Database Systems (ACIIDS)*, 2022.

Publications[1]

- **Anran Hao**, Jian Su, Shuo Sun, Teo Yong Sen, "Soft Syntactic Reinforcement for Neural Event Extraction." *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2025 (accepted).

- **Anran Hao***, Haohan Yuan*, Siu Cheung Hui, Jian Su, "Effective Type Label-based Synergistic Representation Learning for Biomedical Event Trigger Detection." *BMC Bioinformatics*, 2024.

- **Anran Hao**, Anh Tuan Luu, Siu Cheung Hui, Jian Su, "A contrastive learning framework for Event Detection via semantic type prototype representation modelling." *Neurocomputing*, Volume 556, 1 November 2023.

[1]The superscript * indicates joint first authors.

- **Anran Hao**, Siu Cheung Hui, Jian Su, "Semantic Pivoting Model for Effective Event Detection." *In proceedings of the 14th Asian Conference on Intelligent Information and Database Systems (ACIIDS)*, 2022.

Submitted Works

- **Anran Hao**, Shuo Sun, Jian Su, Siu Cheung Hui, Anh Tuan Luu, "Dynamic Task Balancing for Joint Information Extraction." *Submitted to Neurocomputing*, 2024.

Bibliography

[1] Ralph Grishman. Twenty-five years of information extraction. *Natural Language Engineering*, 25(6):677–692, 2019. doi: 10.1017/S1351324919000512. 1, 2, 23

[2] David Ahn. The stages of event extraction. In Branimir Boguraev, Rafael Muñoz, and James Pustejovsky, editors, *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, pages 1–8, Sydney, Australia, July 2006. Association for Computational Linguistics. URL `https://aclanthology.org/W06-0901`. 1, 2, 3, 10, 31

[3] Wei Xiang and Bang Wang. A survey of event extraction from text. *IEEE Access*, 7:173111–173137, 2019. doi: 10.1109/ACCESS.2019.2956831. 1, 2, 24

[4] Xinya Du and Claire Cardie. Document-level event role filler extraction using multi-granularity contextualized encoding. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8010–8020, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.714. URL `https://aclanthology.org/2020.acl-main.714`. 2, 14, 47, 53, 54

[5] Hanzhang Zhou and Kezhi Mao. Document-level event argument extraction by leveraging redundant information and closed boundary loss. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3041–3052, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.222. URL `https://aclanthology.org/2022.naacl-main.222`. 2, 14, 53, 54

[6] Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. Event extraction via dynamic multi-pooling convolutional neural networks. In Chengqing Zong and Michael Strube, editors, *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 167–176, Beijing, China, July 2015. Association

for Computational Linguistics. doi: 10.3115/v1/P15-1017. URL https://aclanthology.org/P15-1017. 3, 10, 13, 14, 23, 31, 32

[7] George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. The automatic content extraction (ACE) program – tasks, data, and evaluation. In Maria Teresa Lino, Maria Francisca Xavier, Fátima Ferreira, Rute Costa, and Raquel Silva, editors, *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal, May 2004. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2004/pdf/5.pdf. 9

[8] Linguistic Data Consortium. *ACE (Automatic Content Extraction) English Annotation Guidelines for Events*, 5.4.3 2005.07.01 edition, 2005. 9

[9] Heng Ji and Ralph Grishman. Refining event extraction through cross-document inference. In Johanna D. Moore, Simone Teufel, James Allan, and Sadaoki Furui, editors, *Proceedings of ACL-08: HLT*, pages 254–262, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL https://aclanthology.org/P08-1030. 10

[10] Thien Huu Nguyen and Ralph Grishman. Event detection and domain adaptation with convolutional neural networks. In Chengqing Zong and Michael Strube, editors, *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 365–371, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-2060. URL https://aclanthology.org/P15-2060. 10

[11] Ziqi Wang, Xiaozhi Wang, Xu Han, Yankai Lin, Lei Hou, Zhiyuan Liu, Peng Li, Juanzi Li, and Jie Zhou. CLEVE: Contrastive Pre-training for Event Extraction. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6283–6297, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.491. URL https://aclanthology.org/2021.acl-long.491. 10, 12, 33

[12] Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. Joint event extraction via recurrent neural networks. In Kevin Knight, Ani Nenkova, and Owen Rambow, editors, *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 300–309, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1034. URL https://aclanthology.org/N16-1034. 10, 13, 14, 37, 42

[13] Shulin Liu, Yubo Chen, Kang Liu, and Jun Zhao. Exploiting argument information to improve event detection via supervised attention mechanisms.

In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1789–1798, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1164. URL `https://aclanthology.org/P17-1164`. 10

[14] Xiao Liu, Zhunchen Luo, and Heyan Huang. Jointly multiple events extraction via attention-based graph information aggregation. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1247–1256, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1156. URL `https://aclanthology.org/D18-1156`. 10, 13, 24, 37, 38

[15] Huazheng Wang, Zhe Gan, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, and Hongning Wang. Adversarial domain adaptation for machine reading comprehension. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2510–2520, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1254. URL `https://aclanthology.org/D19-1254`. 10, 15, 32, 33

[16] Haoran Yan, Xiaolong Jin, Xiangbin Meng, Jiafeng Guo, and Xueqi Cheng. Event detection with multi-order graph convolution and aggregated attention. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5766–5770, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1582. URL `https://aclanthology.org/D19-1582`. 10, 33, 42

[17] Yubo Chen, Shulin Liu, Xiang Zhang, Kang Liu, and Jun Zhao. Automatically labeled data generation for large scale event extraction. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 409–419, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1038. URL `https://aclanthology.org/P17-1038`. 10, 24

[18] Xinya Du and Claire Cardie. Event extraction by answering (almost) natural questions. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 671–683, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.49. URL `https://aclanthology.org/2020.emnlp-main.49`. 10, 13, 32, 43, 51

[19] Yue Zhao, Xiaolong Jin, Yuanzhuo Wang, and Xueqi Cheng. Document embedding enhanced event detection with hierarchical and supervised attention. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 414–419, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2066. URL https://aclanthology.org/P18-2066. 10

[20] Chenwei Zhang, Yaliang Li, Nan Du, Wei Fan, and Philip Yu. Joint slot filling and intent detection via capsule neural networks. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5259–5267, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1519. URL https://aclanthology.org/P19-1519. 10

[21] Jian Liu, Yubo Chen, Kang Liu, and Jun Zhao. Neural cross-lingual event detection with minimal parallel resources. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 738–748, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1068. URL https://aclanthology.org/D19-1068. 10, 42

[22] Ananya Subburathinam, Di Lu, Heng Ji, Jonathan May, Shih-Fu Chang, Avirup Sil, and Clare Voss. Cross-lingual structure transfer for relation and event extraction. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 313–325, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1030. URL https://aclanthology.org/D19-1030. 10

[23] Yaojie Lu, Hongyu Lin, Xianpei Han, and Le Sun. Distilling discrimination and generalization knowledge for event detection via delta-representation learning. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4366–4376, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1429. URL https://aclanthology.org/P19-1429. 10

[24] Jian Liu, Yufeng Chen, and Jinan Xu. Saliency as evidence: Event detection with trigger saliency attribution. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4573–4585, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.313. URL https://aclanthology.org/2022.acl-long.313. 10

[25] Xiao Liu, Heyan Huang, Ge Shi, and Bo Wang. Dynamic prefix-tuning for generative template-based event extraction. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5216–5228, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.358. URL https://aclanthology.org/2022.acl-long.358. 11

[26] Sijia Wang, Mo Yu, Shiyu Chang, Lichao Sun, and Lifu Huang. Query and extract: Refining event extraction as type-oriented binary decoding. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Findings of the Association for Computational Linguistics: ACL 2022*, pages 169–182, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.16. URL https://aclanthology.org/2022.findings-acl.16. 11

[27] Nikolaos Pappas and James Henderson. Gile: A generalized input-label embedding for text classification. *Trans. Assoc. Comput. Linguistics*, 7:139–155, 2019. URL https://transacl.org/ojs/index.php/tacl/article/view/1550. 11

[28] Guoyin Wang, Chunyuan Li, Wenlin Wang, Yizhe Zhang, Dinghan Shen, Xinyuan Zhang, Ricardo Henao, and Lawrence Carin. Joint embedding of words and labels for text classification. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2321–2331, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1216. URL https://aclanthology.org/P18-1216.

[29] Honglun Zhang, Liqiang Xiao, Wenqing Chen, Yongkun Wang, and Yaohui Jin. Multi-task label embedding for text classification. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4545–4553, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1484. URL https://aclanthology.org/D18-1484. 11

[30] Linh The Nguyen, Linh Van Ngo, Khoat Than, and Thien Huu Nguyen. Employing the correspondence of relations and connectives to identify implicit discourse relations via label embeddings. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4201–4207, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1411. URL https://aclanthology.org/P19-1411. 11

[31] Hongming Zhang, Haoyu Wang, and Dan Roth. Zero-shot Label-aware Event Trigger and Argument Classification. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Findings of the Association for Computational*

Linguistics: ACL-IJCNLP 2021, pages 1331–1340, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.114. URL https://aclanthology.org/2021.findings-acl.114. 11

[32] Lifu Huang and Heng Ji. Semi-supervised new event type induction and event detection. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 718–724, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.53. URL https://aclanthology.org/2020.emnlp-main.53. 11, 33

[33] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020. 12, 25

[34] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546 vol. 1, 2005. doi: 10.1109/CVPR.2005.202. 12

[35] Jingfei Du, Edouard Grave, Beliz Gunel, Vishrav Chaudhary, Onur Celebi, Michael Auli, Veselin Stoyanov, and Alexis Conneau. Self-training improves pre-training for natural language understanding. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5408–5418, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.426. URL https://aclanthology.org/2021.naacl-main.426. 12

[36] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding, 2015. URL https://arxiv.org/abs/1511.06452. 12

[37] Daniel Ponsa Vassileios Balntas, Edgar Riba and Krystian Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In Edwin R. Hancock Richard C. Wilson and William A. P. Smith, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 119.1–119.11. BMVA Press, September 2016. ISBN 1-901725-59-6. doi: 10.5244/C.30.119. URL https://dx.doi.org/10.5244/C.30.119. 12

[38] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 1857–1865, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819. 12

[39] Ruihan Zhang, Wei Wei, Xian-Ling Mao, Rui Fang, and Dangyang Chen. HCL-TAT: A hybrid contrastive learning method for few-shot event detection with task-adaptive threshold. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1808–1819, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.130. URL `https://aclanthology.org/2022.findings-emnlp.130`. 12

[40] Senhui Zhang, Tao Ji, Wendi Ji, and Xiaoling Wang. Zero-shot event detection based on ordered contrastive learning and prompt-based prediction. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2572–2580, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-naacl.196. URL `https://aclanthology.org/2022.findings-naacl.196`.

[41] Shunyu Yao, Jian Yang, Xiangqun Lu, and Kai Shuang. Contrastive learning for event extraction. In *2022 The 6th International Conference on Machine Learning and Soft Computing*, ICMLSC 2022, page 167–172, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450387477. doi: 10.1145/3523150.3523176. URL `https://doi.org/10.1145/3523150.3523176`. 12

[42] Wei Xiang and Bang Wang. A survey of event extraction from text. *IEEE Access*, 7:173111–173137, 2019. doi: 10.1109/ACCESS.2019.2956831. 12, 42

[43] Qi Li, Heng Ji, and Liang Huang. Joint event extraction via structured prediction with global features. In Hinrich Schuetze, Pascale Fung, and Massimo Poesio, editors, *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 73–82, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL `https://aclanthology.org/P13-1008`. 12, 13, 15

[44] David McClosky, Mihai Surdeanu, and Christopher Manning. Event extraction as dependency parsing. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, editors, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1626–1635, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL `https://aclanthology.org/P11-1163`. 12

[45] Amir Pouran Ben Veyseh, Tuan Ngo Nguyen, and Thien Huu Nguyen. Graph transformer networks with syntactic and semantic structures for event argument extraction. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3651–3661, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.326. URL `https://aclanthology.org/2020.findings-emnlp.326`. 13, 45

[46] Viet Dac Lai, Tuan Ngo Nguyen, and Thien Huu Nguyen. Event detection: Gate diversity and syntactic importance scores for graph convolution neural networks. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5405–5411, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.435. URL https://aclanthology.org/2020.emnlp-main.435. 13, 33, 42

[47] I-Hung Hsu, Kuan-Hao Huang, Shuning Zhang, Wenxin Cheng, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. TAGPRIME: A unified framework for relational structure extraction. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12917–12932, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.723. URL https://aclanthology.org/2023.acl-long.723. 13, 51

[48] David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. Entity, relation, and event extraction with contextualized span representations. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5784–5789, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1585. URL https://aclanthology.org/D19-1585. 13, 31, 51, 63, 70, 71, 72

[49] Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. A joint neural model for information extraction with global features. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.713. URL https://aclanthology.org/2020.acl-main.713. 13, 15, 43, 51, 60

[50] Zixuan Zhang and Heng Ji. Abstract Meaning Representation guided graph encoding and decoding for joint information extraction. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 39–49, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.4. URL https://aclanthology.org/2021.naacl-main.4. 13, 51

[51] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for*

Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423. 13, 18, 26, 27, 47

[52] I-Hung Hsu, Kuan-Hao Huang, Elizabeth Boschee, Scott Miller, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. DEGREE: A data-efficient generation-based event extraction model. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1890–1908, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.138. URL https://aclanthology.org/2022.naacl-main.138. 13, 51, 81

[53] Kuan-Hao Huang, I-Hung Hsu, Tanmay Parekh, Zhiyu Xie, Zixuan Zhang, Prem Natarajan, Kai-Wei Chang, Nanyun Peng, and Heng Ji. TextEE: Benchmark, reevaluation, reflections, and future challenges in event extraction. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12804–12825, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.760. URL https://aclanthology.org/2024.findings-acl.760/. 13, 14

[54] Jian Liu, Yubo Chen, Kang Liu, Wei Bi, and Xiaojiang Liu. Event extraction as machine reading comprehension. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1641–1651, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.128. URL https://aclanthology.org/2020.emnlp-main.128. 13, 51

[55] Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, and Enhong Chen. Large language models for generative information extraction: A survey. *Frontiers Comput. Sci.*, 18:186357, 2023. URL https://api.semanticscholar.org/CorpusID:266690657. 14

[56] Hao Peng, Xiaozhi Wang, Jianhui Chen, Weikai Li, Yun Peng Qi, Zimu Wang, Zhili Wu, Kaisheng Zeng, Bin Xu, Lei Hou, and Juanzi Li. When does in-context learning fall short and why? a study on specification-heavy tasks. *ArXiv*, abs/2311.08993, 2023. URL https://api.semanticscholar.org/CorpusID:265212914. 14

[57] John Hewitt and Christopher D. Manning. A structural probe for finding syntax in word representations. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language*

Technologies, Volume 1 (Long and Short Papers), pages 4129–4138, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1419. URL https://aclanthology.org/N19-1419. 14, 41, 43, 44, 46

[58] Rowan Hall Maudslay, Josef Valvoda, Tiago Pimentel, Adina Williams, and Ryan Cotterell. A tale of a probe and a parser. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7389–7395, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.659. URL https://aclanthology.org/2020.acl-main.659.

[59] Huayang Li, Lemao Liu, Guoping Huang, and Shuming Shi. On the branching bias of syntax extracted from pre-trained language models. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4473–4478, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.401. URL https://aclanthology.org/2020.findings-emnlp.401.

[60] Benjamin Newman, Kai-Siang Ang, Julia Gong, and John Hewitt. Refining targeted syntactic evaluation of language models. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3710–3723, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.290. URL https://aclanthology.org/2021.naacl-main.290. 14, 43, 44

[61] Hang Yang, Yubo Chen, Kang Liu, Jun Zhao, Zuyu Zhao, and Weijian Sun. Multi-turn and multi-granularity reader for document-level event extraction. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 22(2), dec 2022. ISSN 2375-4699. doi: 10.1145/3542925. URL https://doi.org/10.1145/3542925. 14, 53, 54

[62] Kung-Hsiang Huang, Sam Tang, and Nanyun Peng. Document-level entity-based extraction as template generation. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5257–5269, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.426. URL https://aclanthology.org/2021.emnlp-main.426. 14, 53, 54

[63] Bishan Yang and Tom M. Mitchell. Joint extraction of events and entities within a document context. In Kevin Knight, Ani Nenkova, and Owen

Rambow, editors, *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 289–299, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1033. URL https://aclanthology.org/N16-1033. 15

[64] Trung Minh Nguyen and Thien Huu Nguyen. One for all: Neural joint modeling of entities and events. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6851–6858. AAAI Press, 2019. doi: 10.1609/aaai.v33i01.33016851. URL https://doi.org/10.1609/aaai.v33i01.33016851. 15

[65] Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. A general framework for information extraction using dynamic span graphs. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3036–3046, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1308. URL https://aclanthology.org/N19-1308. 15

[66] Zexuan Zhong and Danqi Chen. A frustratingly easy approach for entity and relation extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online, 2021. Association for Computational Linguistics. 15

[67] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491, 2018. 15, 16, 17, 61

[68] Y. Zhang and Q. Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 34(12):5586–5609, dec 2022. ISSN 1558-2191. doi: 10.1109/TKDE.2021.3070203. 16

[69] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. *Unified language model pre-training for natural language understanding and generation*. Curran Associates Inc., Red Hook, NY, USA, 2019. 16

[70] Sarawoot Kongyoung, Craig Macdonald, and Iadh Ounis. Multi-task learning using dynamic task weighting for conversational question answering. In Jeff Dalton, Aleksandr Chuklin, Julia Kiseleva, and Mikhail Burtsev, editors, *Proceedings of the 5th International Workshop on Search-Oriented Conversational AI (SCAI)*, pages 17–26, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.scai-1.3. URL https://aclanthology.org/2020.scai-1.3. 16

[71] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, 13:281–305, 2012. 16

[72] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, NIPS'11, page 2546–2554, Red Hook, NY, USA, 2011. Curran Associates Inc. ISBN 9781618395993. 16

[73] Gökcen Eraslan, Žiga Avsec, Julien Gagneur, and Fabian J Theis. Deep learning: new computational modelling techniques for genomics. *Nature Reviews Genetics*, 20(7):389–403, 2019. 16

[74] Jörg Fliege and Benar Fux Svaiter. Steepest descent methods for multicriteria optimization. *Mathematical Methods of Operations Research*, 51(3):479–494, 2000. 17, 60, 61, 67

[75] Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qing-Fu Zhang, and Sam Kwong. Pareto multi-task learning. In *Advances in Neural Information Processing Systems*, volume 32, pages 12060–12070. Curran Associates, Inc., 2019. 17, 18, 19, 61, 66, 68

[76] Debabrata Mahapatra and Vaibhav Rajan. Multi-task learning with user preferences: Gradient descent with controlled ascent in pareto optimization. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 6597–6607. PMLR, 13–18 Jul 2020. 17

[77] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. 17, 68

[78] Jean-Antoine Désidéri. Multiple-gradient descent algorithm (mgda) for multiobjective optimization. *Comptes Rendus Mathematique*, 350(5):313–318, 2012. ISSN 1631-073X. doi: https://doi.org/10.1016/j.crma.2012.03.014. 17

[79] Jin-Dong Kim, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, and Jun'ichi Tsujii. Overview of BioNLP'09 shared task on event extraction. In Jun'ichi Tsujii, editor, *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*, pages 1–9, Boulder, Colorado, June 2009. Association for Computational Linguistics. URL `https://aclanthology.org/W09-1401`. 20

[80] Jin-Dong Kim, Sampo Pyysalo, Tomoko Ohta, Robert Bossy, Ngan Nguyen, and Jun'ichi Tsujii. Overview of BioNLP shared task 2011. In Jun'ichi Tsujii, Jin-Dong Kim, and Sampo Pyysalo, editors, *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 1–6, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL `https://aclanthology.org/W11-1801`. 20

[81] Sampo Pyysalo, Filip Ginter, Hans Moen, Tapio Salakoski, and Sophia Ananiadou. Event extraction across multiple levels of biological organization. *Bioinformatics*, 28(18):i575–i581, 2012. URL `https://academic.oup.com/bioinformatics/article/28/18/i575/245396`. 20

[82] Iz Beltagy, Kyle Lo, and Arman Cohan. SciBERT: A pretrained language model for scientific text. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1371. URL `https://aclanthology.org/D19-1371`. 20, 71

[83] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4): 1234–1240, September 2019. ISSN 1367-4811. doi: 10.1093/bioinformatics/btz682. URL `http://dx.doi.org/10.1093/bioinformatics/btz682`. 20

[84] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare*, 3(1):1–23, October 2021. ISSN 2637-8051. doi: 10.1145/3458754. URL `http://dx.doi.org/10.1145/3458754`. 20

[85] Amit Majumder. Multiple features based approach to extract bio-molecular event triggers using conditional random field. *International Journal of Intelligent Systems and Applications*, 4(12):41, 2012. 21

[86] Yijia Zhang, Hongfei Lin, Zhihao Yang, Jian Wang, and Yanpeng Li. Biomolecular event trigger detection using neighborhood hash features. *Journal of theoretical biology*, 318:22–28, 2013.

[87] Jari Björne and Tapio Salakoski. Tees 2.1: Automated annotation scheme learning in the bionlp 2013 shared task. In *Proceedings of the BioNLP shared task 2013 workshop*, pages 16–25, 2013.

[88] Deyu Zhou, Dayou Zhong, and Yulan He. Event trigger identification for biomedical events extraction using domain knowledge. *Bioinformatics*, 30 (11):1587–1594, 2014. 81

[89] Bishan Yang and Tom Mitchell. Joint extraction of events and entities within a document context. *arXiv preprint arXiv:1609.03632*, 2016. 21

[90] Jari Björne and Tapio Salakoski. Tees 2.2: biomedical event extraction for diverse corpora. *BMC bioinformatics*, 16(16):1–20, 2015. 21

[91] Jari Björne and Tapio Salakoski. Biomedical event extraction using convolutional neural networks and dependency parsing. In *Proceedings of the BioNLP 2018 workshop*, pages 98–108, 2018. 21, 81

[92] Yan Wang, Jian Wang, Hongfei Lin, Xiwei Tang, Shaowu Zhang, and Lishuang Li. Bidirectional long short-term memory with crf for detecting biomedical event trigger in fasttext semantic space. *BMC bioinformatics*, 19: 59–66, 2018. 21, 81

[93] Yufeng Diao, Hongfei Lin, Liang Yang, Xiaochao Fan, Di Wu, Zhihao Yang, Jian Wang, and Kan Xu. Fbsn: A hybrid fine-grained neural network for biomedical event trigger identification. *Neurocomputing*, 381:105–112, 2020.

[94] Hai-Long Trieu, Thy Thy Tran, Khoa NA Duong, Anh Nguyen, Makoto Miwa, and Sophia Ananiadou. Deepeventmine: end-to-end neural nested event extraction from biomedical texts. *Bioinformatics*, 36(19):4910–4917, 2020. 81

[95] Hao Wei, Ai Zhou, Yijia Zhang, Fei Chen, Wen Qu, and Mingyu Lu. Biomedical event trigger extraction based on multi-layer residual bilstm and contextualized word representations. *International Journal of Machine Learning and Cybernetics*, pages 1–13, 2022. 21

[96] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017. doi: 10.1162/tacl_a_00051. URL `https://aclanthology.org/Q17-1010`. 21

[97] Kung-Hsiang Huang, Mu Yang, and Nanyun Peng. Biomedical event extraction with hierarchical knowledge graphs. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1277–1285, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.114. URL `https://aclanthology.org/2020.findings-emnlp.114`. 21

[98] Shasha Liao and Ralph Grishman. Using document level cross-event inference to improve event extraction. In Jan Hajič, Sandra Carberry, Stephen Clark, and Joakim Nivre, editors, *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 789–797, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL `https://aclanthology.org/P10-1081`. 23

[99] Shulin Liu, Kang Liu, Shizhu He, and Jun Zhao. A probabilistic soft logic based approach to exploiting latent and global information in event classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1), Mar. 2016. doi: 10.1609/aaai.v30i1.10375. URL `https://ojs.aaai.org/index.php/AAAI/article/view/10375`. 23

[100] Xiaozhi Wang, Ziqi Wang, Xu Han, Wangyi Jiang, Rong Han, Zhiyuan Liu, Juanzi Li, Peng Li, Yankai Lin, and Jie Zhou. MAVEN: A Massive General Domain Event Detection Dataset. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1652–1671, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.129. URL `https://aclanthology.org/2020.emnlp-main.129`. 24, 31, 32, 33, 51

[101] Yubo Chen, Hang Yang, Kang Liu, Jun Zhao, and Yantao Jia. Collective event detection via a hierarchical and bias tagging networks with gated multi-level attention mechanisms. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1267–1276, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1158. URL `https://aclanthology.org/D18-1158`. 24, 37, 38

[102] Thien Huu Nguyen, Nicolas Fauceglia, Mariano Rodriguez Muro, Oktie Hassanzadeh, Alfio Massimiliano Gliozzo, and Mohammad Sadoghi. Joint learning of local and global features for entity linking via neural networks. In Yuji Matsumoto and Rashmi Prasad, editors, *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2310–2320, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee. URL `https://aclanthology.org/C16-1218`. 24, 38

[103] Lance Ramshaw and Mitch Marcus. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*, 1995. URL `https://aclanthology.org/W95-0107`. 29

[104] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1558607781. 30

[105] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 32

[106] Xiaozhi Wang, Ziqi Wang, Xu Han, Zhiyuan Liu, Juanzi Li, Peng Li, Maosong Sun, Jie Zhou, and Xiang Ren. HMEAE: Hierarchical modular event argument extraction. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5777–5783,

Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1584. URL https://aclanthology.org/D19-1584. 32, 51

[107] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. URL http://arxiv.org/abs/1412.6980. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015. 32, 51

[108] Meihan Tong, Shuai Wang, Yixin Cao, Bin Xu, Juanzi Li, Lei Hou, and Tat-Seng Chua. Image enhanced event detection in news articles. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9040–9047. AAAI Press, 2020. URL https://aaai.org/ojs/index.php/AAAI/article/view/6437. 33

[109] Dongfang Lou, Zhilin Liao, Shumin Deng, Ningyu Zhang, and Huajun Chen. MLBiNet: A cross-sentence collective event detection network. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4829–4839, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.373. URL https://aclanthology.org/2021.acl-long.373. 33

[110] Devendra Sachan, Yuhao Zhang, Peng Qi, and William L. Hamilton. Do syntax trees help pre-trained transformers extract information? In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2647–2661, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.228. URL https://aclanthology.org/2021.eacl-main.228. 43

[111] Sebastian Ruder, Matthew E. Peters, Swabha Swayamdipta, and Thomas Wolf. Transfer learning in natural language processing. In Anoop Sarkar and Michael Strube, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*, pages 15–18, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-5004. URL https://aclanthology.org/N19-5004. 43

[112] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993. URL https://aclanthology.org/J93-2004. 46

[113] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019. 47

[114] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. 47

[115] Kuan-Hao Huang, I-Hung Hsu, Tanmay Parekh, Zhiyu Xie, Zixuan Zhang, Premkumar Natarajan, Kai-Wei Chang, Nanyun Peng, and Heng Ji. Textee: Benchmark, reevaluation, reflections, and future challenges in event extraction, 2024. 49

[116] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`. 51

[117] Jiangang Bai, Yujing Wang, Yiren Chen, Yaming Yang, Jing Bai, Jing Yu, and Yunhai Tong. Syntax-BERT: Improving pre-trained transformers with syntax trees. In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3011–3020, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.262. URL `https://aclanthology.org/2021.eacl-main.262`. 51

[118] Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017. 51

[119] Beth M. Sundheim. Overview of the fourth Message Understanding Evaluation and Conference. In *Fourth Message Understanding Conference (MUC-4): Proceedings of a Conference Held in McLean, Virginia, June 16-18, 1992*, 1992. URL `https://aclanthology.org/M92-1001`. 53

[120] Ruihong Huang and Ellen Riloff. Modeling textual cohesion for event extraction. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, page 1664–1670. AAAI Press, 2012. 53, 54

[121] Xinya Du, Alexander Rush, and Claire Cardie. GRIT: Generative role-filler transformers for document-level event entity extraction. In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 634–644, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.52. URL `https://aclanthology.org/2021.eacl-main.52`. 53

[122] Zhaohui Yan, Songlin Yang, Wei Liu, and Kewei Tu. Joint entity and relation extraction with span pruning and hypergraph neural networks. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7512–7526, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.467. URL https://aclanthology.org/2023.emnlp-main.467. 60, 63

[123] Jue Wang and Wei Lu. Two are better than one: Joint entity and relation extraction with table-sequence encoders. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1706–1721, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.133. URL https://aclanthology.org/2020.emnlp-main.133.

[124] Minh Van Nguyen, Bonan Min, Franck Dernoncourt, and Thien Nguyen. Joint extraction of entities, relations, and events via modeling inter-instance and inter-label dependencies. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4363–4374, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.324. URL https://aclanthology.org/2022.naacl-main.324. 63

[125] Yanguang Chen, Yuanyuan Sun, Zhihao Yang, and Hongfei Lin. Joint entity and relation extraction for legal documents with legal feature enhancement. In Donia Scott, Nuria Bel, and Chengqing Zong, editors, *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1561–1571, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.137. URL https://aclanthology.org/2020.coling-main.137. 63

[126] Yuren Mao, Zekai Wang, Weiwei Liu, Xuemin Lin, and Pengtao Xie. MetaWeighting: Learning to weight tasks in multi-task learning. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3436–3448, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.271. URL https://aclanthology.org/2022.findings-acl.271. 64

[127] Stephen Boyd and LievenEditor Vandenberghe. *Convex Optimization*, page 216. Cambridge University Press, 2004. 68

[128] Yuka Tateisi, Akane Yakushiji, Tomoko Ohta, and Jun'ichi Tsujii. Syntax annotation for the GENIA corpus. In *Companion Volume to the Proceedings of Conference including Posters/Demos and tutorial abstracts*, 2005. URL https://aclanthology.org/I05-2038. 79

[129] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Shruti Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019. 87

[130] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 87

[131] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. 87

[132] Lei Luo and Yajing Xu. Context-aware prompt for generation-based event argument extraction with diffusion models. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, page 1717–1725, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701245. doi: 10.1145/3583780.3614820. URL https://doi.org/10.1145/3583780.3614820. 88

[133] Xingyao Wang, Sha Li, and Heng Ji. Code4Struct: Code generation for few-shot event structure prediction. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3640–3663, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.202. URL https://aclanthology.org/2023.acl-long.202. 88

[134] Zhihan Zhang, Wenhao Yu, Mengxia Yu, Zhichun Guo, and Meng Jiang. A survey of multi-task learning in natural language processing: Regarding task relatedness and training methods. In Andreas Vlachos and Isabelle Augenstein, editors, *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 943–956, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.66. URL https://aclanthology.org/2023.eacl-main.66. 88

[135] Wenqing Chen, Jidong Tian, Liqiang Xiao, Hao He, and Yaohui Jin. Exploring logically dependent multi-task learning with causal inference. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2213–2225, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.173. URL https://aclanthology.org/2020.emnlp-main.173. 88

[136] Yubo Ma, Zehao Wang, Yixin Cao, and Aixin Sun. Few-shot event detection: An empirical study and a unified view. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11211–11236, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.628. URL https://aclanthology.org/2023.acl-long.628. 89