LOW-RESOURCE EVENT EXTRACTION

by

VIET DAC LAI

A DISSERTATION

Presented to the Department of Computer Science
and the Division of Graduate Studies of the University of Oregon
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

September 2023

DISSERTATION APPROVAL PAGE

Student: Viet Dac Lai

Title: Low-Resource Event Extraction

This dissertation has been accepted and approved in partial fulfillment of the requirements for the Doctor of Philosophy degree in the Department of Computer Science by:

Thien Huu Nguyen	Chair
Daniel Lowd	Core Member
Humphrey Shi	Core Member
Gabriela Pérez Báez	Institutional Representative

and

Krista Chronister	Vice Provost for Graduate Studies

Original approval signatures are on file with the University of Oregon Division of Graduate Studies.

Degree awarded September 2023

DISSERTATION ABSTRACT

Viet Dac Lai

Doctor of Philosophy

Department of Computer Science

September 2023

Title: Low-Resource Event Extraction

The last decade has seen the extraordinary evolution of deep learning in natural language processing leading to the rapid deployment of many natural language processing applications. However, the field of event extraction did not witness a parallel success story due to the inherent challenges associated with its scalability. The task itself is much more complex than other NLP tasks due to the dependency among its subtasks. This interlocking system of tasks requires a full adaptation whenever one attempts to scale to another domain or language, which is too expensive to scale to thousands of domains and languages. This dissertation introduces a holistic method for expanding event extraction to other domains and languages within the limited available tools and resources. First, this study focuses on designing neural network architecture that enables the integration of external syntactic and graph features as well as external knowledge bases to enrich the hidden representations of the events. Second, this study presents network architecture and training methods for efficient learning under minimal supervision. Third, we created brand new multilingual corpora for event relation extraction to facilitate the research of event extraction in low-resource languages. We also introduce a language-agnostic method to tackle multilingual event relation extraction. Our extensive experiment shows the effectiveness of these methods

which will significantly speed up the advance of the event extraction field. We anticipate that this research will stimulate the growth of the event detection field in unexplored domains and languages, ultimately leading to the expansion of language technologies into a more extensive range of diaspora.

This dissertation includes both previously published and co-authored material.

CURRICULUM VITAE

NAME OF AUTHOR: Viet Dac Lai

GRADUATE AND UNDERGRADUATE SCHOOLS ATTENDED:

University of Oregon, Eugene, Oregon, USA
Japan Advanced Institute of Science and Technology, Nomi, Ishikawa, Japan
Posts and Telecommunications Institute of Technology, Hanoi, Vietnam

DEGREES AWARDED:

Doctor of Philosophy, Computer Science, 2023, University of Oregon
Master of Science, Computer Science, 2018, Japan Advanced Institute of Science and Technology
Bachelor of Arts, Information Technology, 2016, Posts and Telecommunications Institute of Technology

AREAS OF SPECIAL INTEREST:

Natural Language Processing
Information Extraction
Transfer Learning
Low Resource Learning

PROFESSIONAL EXPERIENCE:

Teaching Assistant, Department of Computer Science, University of Oregon
Research Scientist Intern, Adobe Research
Reviewer: ACL Rolling Review, ACL, NAACL, Neurocomputing.

GRANTS, AWARDS AND HONORS:

Erwin & Gertrude Juilfs Scholarship
Dept. of Computer Science, University of Oregon, 2022
Adobe Research Fellowship, Adobe Inc., 2022

Best Graduate Teaching Assistant
Dept. of Computer Science, University of Oregon, 2021

PUBLICATIONS:

Viet Dac Lai, Tuan Ngo Nguyen, and Thien Huu Nguyen (2020). Event Detection: Gate Diversity and Syntactic Importance Scores for Graph Convolution Neural Networks. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 5405-5411).

Viet Dac Lai, Minh Van Nguyen, Thien Huu Nguyen, and Franck Dernoncourt (2021). Graph learning regularization and transfer learning for few-shot event detection. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 2172-2176).

Viet Dac Lai, Franck Dernoncourt, and Thien Huu Nguyen (2021). Learning Prototype Representations Across Few-Shot Tasks for Event Detection. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (pp. 5270-5277).

Viet Dac Lai, Amir Pouran Ben Veyseh, Minh Van Nguyen, Franck Dernoncourt, and Thien Huu Nguyen (2022). MECI: A multilingual dataset for event causality identification. *In Proceedings of the 29th International Conference on Computational Linguistics*, (pp. 2346-2356).

Viet Dac Lai, Hieu Man, Linh Ngo, Franck Dernoncourt, and Thien Huu Nguyen (2022). Multilingual SubEvent Relation Extraction: A Novel Dataset and Structure Induction Method. *Findings of the Association for Computational Linguistics: EMNLP 2022*, (pp. 5559-5570).

Viet Dac Lai, Abel Salinas, Hao Tan, Trung Bui, Quan Tran, Seunghyun Yoon, Hanieh Deilamsalehy, Franck Dernoncourt, Thien Huu Nguyen (2023, August). Boosting Punctuation Restoration with Data Generation and Reinforcement Learning. *In INTERSPEECH 2023, 24th Annual Conference of the International Speech Communication Association, 2023*.

Viet Dac Lai, Amir Pouran Ben Veyseh, Franck Dernoncourt, and Thien Huu Nguyen (2022, July). Behancepr: A punctuation restoration dataset for livestreaming video transcript. *In Findings of the Association for Computational Linguistics: NAACL 2022* (pp. 1943-1951).

Viet Dac Lai, Minh Van Nguyen, Heidi Kaufman, and Nguyen, T. H. (2021, August). Event extraction from historical texts: A new dataset for black rebellions. *In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021* (pp. 2390-2400).

Viet Dac Lai, Franck Dernoncourt, and Thien Huu Nguyen (2020, July). Extensively Matching for Few-shot Learning Event Detection. *In Proceedings of the First Joint Workshop on Narrative Understanding, Storylines, and Events* (pp. 38-45).

Viet Dac Lai, Franck Dernoncourt, and Thien Huu Nguyen (2020, May). Exploiting the matching information in the support set for few shot event classification. *In Advances in Knowledge Discovery and Data Mining: 24th Pacific-Asia Conference*, PAKDD 2020, Singapore, May 11–14, 2020, Proceedings, Part II 24 (pp. 233-245). Springer International Publishing.

Viet Dac Lai and Thien Huu Nguyen (2019). Extending Event Detection to New Types with Learning from Keywords. *In Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)* (pp. 243-248).

ACKNOWLEDGEMENTS

My heartfelt thanks go to my advisor, Prof. Thien Huu Nguyen, for presenting me with the opportunity to join the UONLP group. His detailed guidance and immense support have been key drivers in propelling me to this significant milestone in my career. I am extremely grateful to Prof. Daniel Lowd and Prof. Humphrey Shi for their unwavering support and guidance throughout my Ph.D. journey. Additionally, I extend my heartfelt appreciation to Prof. Gabriela Pérez Báez for her contribution as a member of my dissertation. Their invaluable presence as committee members has been instrumental in shaping my academic and research pursuits. I am truly grateful for the expertise and insights they have shared, which have greatly enriched my educational experience.

I express my gratitude to Dr. Franck Dernoncourt, my mentor at Adobe Research, for his significant contributions to my research in terms of both guidance and funding support. His unwavering assistance has played a crucial role in the advancement of my research endeavors.

I extend my gratitude to my exceptional colleagues at the UONLP group for their priceless experiences and teamwork. They include but are not limited to, Amir Pouran Ben Veyseh and Minh Van Nguyen. My Ph.D. journey has been tremendously enriched by their contributions. Furthermore, I can't overlook the unconditional support provided by my peers, Zayd Hammoudeh, Steven Walton, and Yimin Chen, who have truly elevated my doctoral experience.

I extend my sincere appreciation to each faculty and administrative personnel I had the good fortune to work alongside during my time here. I am profoundly grateful for Prof. Hank Childs, whose unfaltering support and consistent

encouragement throughout my five-year Ph.D. journey have been instrumental.
I would also like to acknowledge Dr. Kathleen Freeman and Phil Colbert whose mentorship and shared experiences within UO's teaching environment have greatly enhanced my learning.

To my beloved family.

TABLE OF CONTENTS

Chapter Page

LIST OF FIGURES

Figure Page

LIST OF TABLES

Table Page

Table Page

CHAPTER I

INTRODUCTION

1.1 Introduction

Event Extraction (EE) is an essential task in Information Extraction (IE) in Natural Language Processing (NLP). An event is an occurrence of an activity that happens at a particular time and place, or it might be described as a change of state (LDC, 2005). The main task of event extraction is to detect events in the text (i.e., event detection) and then sort them into some classes of interest (i.e., event classification). The second task involves detecting the event participants (i.e., argument extraction) and their attributes (e.g., argument role labeling). In short, event extraction structures the unstructured text by answering the *WH* questions of an event (i.e., what, who, when, where, why, and how).

Event extraction plays a vital role in various natural language processing applications. For instance, the extracted event can be used to construct knowledge bases on which people can perform logical queries easily (Ge et al., 2018). Many domains can benefit from the development of event extraction research. In the biomedical domain, event extraction can be used to extract interaction between biomolecules (e.g., protein-protein interactions) that have been described in the biomedical literature (Kim, Ohta, Pyysalo, Kano, & Tsujii, 2009). In the economic domain, events reported on social media and social networks can be used for measuring socio-economic indicators (Min & Zhao, 2019). Recently, event extraction has been adopted in many other domains such as literature (Sims, Park, & Bamman, 2019), cyber security (Man Duc Trong, Trong Le, Pouran Ben Veyseh, Nguyen, & Nguyen, 2020), history (Sprugnoli & Tonelli, 2019), and humanity (V. D. Lai, Nguyen, Kaufman, & Nguyen, 2021).

It closely connects with other natural language processing tasks such as named entity recognition (NER), entity linking (EL), and dependency parsing. Although these tasks can boost the development of event extraction (McClosky, Surdeanu, & Manning, 2011), they might have an inverse impact on the performance of the event extraction systems (Y. Zhang, Qi, & Manning, 2018), depending on how the output of these tasks is exploited.

Even though event extraction has been studied for decades, it is still a very challenging task. To perform the event extraction, a system needs to understand the text's semantics and ambiguity and organize the extracted information into structures (LDC, 2005). Lacking training data is also a fundamental problem in expanding event extraction to a new domain or a new language because the traditional classification model requires a large amount of training data (L. Huang et al., 2018). Therefore, extracting events with a substantially small amount of training data is a new and challenging problem.

There has been a great interest in studying event extraction in the last two decades. The majority of the studies have focused on supervised learning for a few domains and the English language, while little attention was paid to other essential domains and the majority of human languages. In this dissertation, we aim to extend event extraction to a broader set of domains and languages. We investigate methods in representation learning, transfer learning, and multilingual learning.

The rest of the dissertation is organized as follows:

- Chapter I presents the definition of the subtasks of event extraction and a literature review of event extraction with a focus on low-resource event extraction.

– Chapter II presents our first work in improving the event extraction models with a novel gating mechanism and a method to inject external syntactic features into the models that are based on graph convolutional neural networks.

– After that, Chapter III steers our focus toward low-resource event detection besides the traditional supervised learning setting. This chapter presents our successful attempt to transfer knowledge from an existing knowledge base of a different task to enrich the representation of the ED model. We also present a new training signal to regularize the representational learning that is based on a graph convolutional neural network.

– Then, Chapter IV fully directs the attention to few-shot learning for ED. We addressed the noise and bias issues of the episodical training setting in few-shot learning for ED by proposing a method to induce a better class-representational prototype. This leads to a significant improvement in the few-shot learning performance while requiring no additional training data during the inference time.

– Chapter V and VI present the first work for multilingual event relation extraction. In these two chapters, we introduce two new corpora for multilingual event relation extraction on causality and subevent relations, respectively.

– Moreover, Chapter VI presents a novel method to utilize optimal transport for selecting the related context in a long document for the event relation extraction task.

– In conclusion, Chapter VII finalizes the dissertation and outlines our future areas of interest for further exploration.

This dissertation contains materials from published and co-authored papers. We acknowledge all the co-authors: Tuan Ngo Nguyen, Thien Huu Nguyen, Minh Van Nguyen, Franck Dernoncourt, Amir Pouran Ben Veyseh, Hieu Man, and Linh Ngo.

1.2 Subtasks

Event extraction aims to detect the appearance of event structure in the text (e.g., sentence, document). This structure includes the event trigger and its related information such as event arguments (e.g., participants, time, location), event argument roles, and event-event relations (e.g., causality, hierarchy, coreference). Event structures are commonly predefined to show the relationship between the event triggers and entities, such as participants and their relations to the event.

ACE-2005 (LDC, 2005) defines an event ontology whose terminologies have been widely used in event extraction:

– An **event extent** is a sentence within which an event is expressed.

– An **event trigger** is a word or phrase that most clearly expresses the event's occurrence. In many cases, the event trigger is the sentence's main verb expressing the event.

– **Event's participants** are the entities that are involved in that event.

– **Event arguments** are entities that are part of the event. They include participants and attributes.

– An **argument role** is the relationship between an event and its arguments.

Based on these terminologies, Ahn (2006) proposes to divide the event extraction into four sub-tasks: trigger detection, trigger classification, argument detection, and argument classification. These subtasks can be done either separately or jointly. Table 1 demonstrates an ideal output that an event extraction system must accomplish given the following sentence.

Earlier documents in the case have included embarrassing details about perks ***Welch*** *received as part of* ***his*** *retirement package from* ***GE*** *at a time when corporate scandals were sparking outrage.*

Trigger	retirement
Event type	Personnel:End-Position
Person-Arg	Welch
Entity-Arg	GE
Position-Arg	-
Time-Arg	-
Place-Arg	-

Table 1. A sample in ACE-05 dataset.

Recently, there has been a great interest in understanding the relation between events in a document. Four particular event-event relations that are concerned the most are **causal, temporal, subevent**, and **coreference** relations. As such, extracting these relations are more and more studied together with the original four main tasks of EE. The following sentence shows a series of events which are marked in bold:

"A massive **quake** *struck off Aceh in 2004, sparking a* **tsunami**.*"*

In this example, an event relation extraction system should mark the causal relation that the *"quake"* caused the *"tsunami"*, signaled by the word *"sparking"*.

This problem is challenging because of the ambiguity of human languages W. Lu and Nguyen (2018) that requires the understanding of not only the true semantics of the specific activities mentioned in the text but also their semantical relations between events and entities (the event argument extraction task), and pairs of events (event relation extraction task). It is important to note that effective models for event extraction require an appropriate understanding of input texts beyond language syntax (or syntactic features), characterizing contextual semantics and relations as the key information that should be inferred from the input text to guarantee successful predictions. In addition, such semantic information can involve explicit or implicit reasoning from the input text where relevant background knowledge is necessary to secure strong performance.

Throughout this dissertation, we will include materials from prior work that refers to different text granularity. The following table shows our definitions, particularly for English. The definitions of granularity such as word, word-piece, character, and token might be different from language to language. Some of them might not exist or use interchangeably. So, when adapting to another language, those terms should be adapted accordingly.

1.3 Corpora

The development of event extraction was mainly promoted by the availability of data offered by public evaluation programs such as Message Understanding Conference (MUC), Automatic Content Extraction (ACE), and Knowledge Base Population (TAC-KBP).

Automatic Content Extraction (ACE-2005) is the most widely used corpus in event extraction for English, Arabic, and Chinese. It annotates entities, events, relations, and time (LDC, 2005). There are 7 categories of entities in ACE-

Sentence	A sentence in this dissertation is defined as a conventional sentence that gives a complete meaning. It ends with a period, a question mark, or an exclamation mark.
Document	A document refers to a sequence of contiguous sentences. In this dissertation, a document is not necessary to be a full/complete article/essay. It can be a single paragraph or multiple paragraphs.
Word	A word is a text unit that is separated by white space. Word is usually used in early work in NLP such as Word2VecMikolov, Chen, Corrado, and Dean (2013) and GLoVe Pennington, Socher, and Manning (2014)
Word piece	Word piece is a segmentation of a word after a word is split into smaller units. A tokenizer is an algorithm that split words into word pieces. Common word-piece tokenizers are WordPiece Y. Wu et al. (2016) and Byte Pair Encoding Radford, Narasimhan, Salimans, and Sutskever (2018)
Token	A token refers to the primitive unit that the model consumes. It can refer to a word, a word piece, or a character depending on the model being used.

Table 2. Text granularity in this dissertation.

2005, i.e., person, organization, location, geopolitical entity, facility, vehicle, and weapon. The ACE-2005 defines 8 event types and 33 event subtypes as presented in table 3. This dataset annotates 599 documents from various sources, e.g., weblogs, broadcast news, newsgroups, and broadcast conversation.

TAC-KBP datasets aim to promote extracting information from unstructured text that fits the knowledge base. The dataset includes the annotation for event detection, event coreference, event linking, argument extraction, and argument linking (Ellis et al., 2015). The event taxonomy in TAC-KBP is mainly derived from ACE-2005, with 9 event types and 38 event subtypes. This dataset contains 360 documents, of which 158 documents are used for training and 202 for testing. The TAC-KBP 2015 contains documents for English only (Ellis et al., 2015), whereas TAC-KBP 2016 includes Chinese and Spanish documents (Ji, Nothman, Dang, & Hub, 2016).

Event type	Event subtype
Life	Be-born, Marry, Divorce, Injure, Die
Movement	Transport
Transaction	Transfer-Ownership, Transfer-Money
Business	Start-Org, Merge-Org, Declare-Bankruptcy, End-Org
Conflict	Attack, Demonstrate
Contact	Meet, Phone-Write
Personnel	Start-Position, End-Position, Nominate, Elect
Justice	Arrest-Jail, Release-Parole, Trial-Hearing, Charge-Indict, Sue, Convict, Sentence, Fine, Execute, Extradite, Acquit, Appeal, Pardon

Table 3. A full list of event types and event subtypes in ACE-2005.

Many corpora for specific domains are available for public use. **MUC** corpus annotates events for domains such as fleet operation, terrorism, and semiconductor production (Grishman & Sundheim, 1996). The **GENIA** is an event detection corpus for the biomedical domain. It is compiled from scientific documents from PubMed by the BioNLP Shared Task (Kim et al., 2009). **TimeBank** annotates 183 English news articles with event, temporal annotations, and their links (Pustejovsky, Hanks, et al., 2003). Recently, event detection has expanded to many other fields such as **CASIE** and **CyberED** for cyber-security (Man Duc Trong et al., 2020; Satyapanich, Ferraro, & Finin, 2020), **Litbank** for literature (Sims et al., 2019), and music (Ding, Song, Qin, & LIU, 2011). However, these corpora are both small in the number of data samples and close in terms of the domain. Consequently, this limits the ability of the pre-trained models to perform tasks in a new domain in real applications.

The above corpora only annotate event extraction at the sentence level. There have been some studies that annotate events at a higher level such as paragraph-level Ebner, Xia, Culkin, Rawlins, and Van Durme (2020) or document-level Xu, Liu, Li, and Chang (2021).

On the other hand, a general-domain dataset for event detection is a good fit for real applications because it offers a much more comprehensive range of domains and topics. However, manually creating a large-scale general-domain dataset for ED is too costly to anyone ever attempt. Instead, general-domain datasets for event detection have been produced at a large scale by exploiting a knowledge base and unlabeled text. Distant supervision and learning models are the two main methods employed to generate large-scale ED datasets.

Distant supervision (Mintz, Bills, Snow, & Jurafsky, 2009) is the most widely use with facts derived from existing knowledge base such as WordNet (Miller, 1995), FrameNet (Baker, Fillmore, & Lowe, 1998), and Freebase (Bollacker, Evans, Paritosh, Sturge, & Taylor, 2008). Y. Chen, Liu, Zhang, Liu, and Zhao (2017) proposes an approach to align key arguments of an event by using Freebase. Then these arguments are used to detect the event and its trigger word automatically. The data is further denoised by using FrameNet (Baker et al., 1998). Similarly, (X. Wang, Wang, et al., 2020) constructs the **MAVEN** dataset from Wikipedia text and FrameNet. This dataset also offers a tree-like event schema structure rooted in the word sense hierarchy in FrameNet. Similarly, (Le & Nguyen, 2021) creates **FedSemcor** from WordNet and Word Sense Disambiguation dataset. A subset of WordNet synsets that are more likely eventive is collected and grouped into event detection classes with similar meanings. The Semcor is a word sense disambiguation dataset whose tokens are labeled by WordNet synsets. To create the event detection, the text from the Semcor dataset is realigned with the collected ED classes.

Table 4 presents a summary of the existing event extraction dataset for ED.

1.4 Supervised Learning Models

1.4.1 Feature-based models.

In the early stage of event extraction, most methods utilize a large set of features (i.e., feature engineering) for statistical classifiers. The features can be derived from constituent parser (Ahn, 2006), dependency parser (Ahn, 2006), POS taggers, unsupervised topic features (Liao & Grishman, 2010), and contextual features (Patwardhan & Riloff, 2009). These models employ statistical models such as nearest neighbor (Ahn, 2006), maximum-entropy classifier (Liao & Grishman, 2010), and conditional random field (Majumder & Ekbal, 2015).

Ahn (2006) employed a rich feature set of lexical, dependency, and entity features. The lexical features include the word and its lemma, lowercase, and Part-of-Speech (POS) tag. The dependency features include the depth of the word in the dependency tree, the dependency relation of the trigger, and the POS of the connected nodes. The context features include left/right contexts, such as lowercase, POS tag, and entity type. The entity features include the number of dependants, labels, constituent headwords, the number of entities along a dependency path, and the path length to the closest entity.

Ji and Grishman (2008) further introduced cross-sentence and cross-document rules to mandate the consistencies of the classification of triggers and their arguments in a document. In particular, they include (1) the consistency of word sense across sentences in related documents and (2) the consistency of roles and entity types for different mentions of the related events.

Patwardhan and Riloff (2009) suggest using contextual features such as the lexical head of the candidate, the semantic class of the lexical head, lexico-semantic pattern surrounding the candidate. This information provides rich contextual

features of the words surrounding the candidate and its lexical-connected words, which provides some signal for the success of convolutional neural networks and graph convolutional neural networks based on the dependency graph in recent studies.

Liao and Grishman (2010) shows that global topic features can help improve EE performance on test data, especially for a balanced corpus. The unsupervised topic model trained on large untagged corpus can provide underlying relations between event and entity types. Therefore, it can reduce the bias introduced in an imbalanced corpus (e.g., ACE-2005 dataset).

Majumder and Ekbal (2015) extracts various features for biomedical event extraction, such as dependency path and distance to the nearest protein entity. Since the terminologies in the biomedical domain follow some particular rules, the suffix-prefix of words provides substantial semantic information about the terms.

Even though tremendous effort has been poured into feature engineering, feature-based models with statistical classifiers hinder the application of event extraction models in practical situations for two reasons. The first reason is the need for the manual design of the feature set, which requires research expertise in both linguistics and the target-specific domain. Second, since feature extraction tools are imperfect, their incorrect extracted features can harm the statistical models. Hence, a model which can automatically learn would significantly boost the application of event extraction.

1.4.2 Neural-based models.

As mentioned in the previous section, crafting a diverse set of lexical, syntactic, semantic, and topic features require both linguistic and domain expertise. This might hinder the adaptability of the model to real applications where expertise

is scarce. Therefore, instead of manually designing linguistic features, automatically extracting features is more practical in virtually every NLP task. Hence, it can revolutionize the common practice of NLP studies. Toward this end, the deep neural network is the perfect match because of its ability to capture features from text automatically.

Deep neural networks employing multiple layers of a large number of artificial neurons have been adapted to various classification and generation tasks. In an artificial neural network, a layer takes input from the output of the lower layer and transforms it into a more abstract representation with two exceptions. The lowest layer takes input as a vector generated from the data sample. The highest layer usually outputs a score for each of the classification classes. These scores are used for the prediction of the label.

1.4.2.1 Distributed word embedding.

Distributed word embedding is one of the most impactful tools for most NLP tasks, including event extraction. Word embedding plays a vital role in transitioning from feature-based to neural-based modeling. The representation obtained from word embedding captures a rich set of syntactic features, semantic features, and knowledge learned from a large amount of text (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013).

Technically, distributed word embedding is a matrix that can be viewed as a list of low-dimensional continuous float vectors (Bengio, Ducharme, Vincent, & Jauvin, 2003). Word embedding maps a word into a single vector within its dictionary. Hence, a sentence can be encoded into a list of vectors. These vectors are fed into the neural network. Among tens of variants, Word2Vec (Mikolov, Sutskever, et al., 2013) and GloVe (Pennington et al., 2014) are the most popular word embeddings. These word embeddings were then called context-free embedding

to distinguish against contextualized word embedding, which was invented a few years after context-free word embedding.

Contextualized word embedding is one of the greatest inventions in the field of NLP recently. Contrary to context-free word embedding, contextualized embedding encodes the word in a sentence based on the context presented in the text (Peters et al., 2018). In addition, the contextualized embeddings are usually trained on a large text corpus. Hence, its embedding encodes a substantial amount of knowledge from the text. These lead to the improvement of virtually every model in NLP. There have been many variants of contextualized word embedding for general English text, e.g., BERT (Devlin, Chang, Lee, & Toutanova, 2019), RoBERTa (Y. Liu et al., 2019), multi-lingual text, e.g., mBERT (Devlin et al., 2019), XLM-RoBERTa (Ruder, Søgaard, & Vulić, 2019), scientific document SciBERT (Beltagy, Lo, & Cohan, 2019), and text generation, e.g., GPT2 (Radford et al., 2019).

1.4.2.2 Convolutional Neural Networks.

T. H. Nguyen and Grishman (2015) employed a convolutional neural network, inspired by CNNs in computer vision (LeCun, Bottou, Bengio, & Haffner, 1998) and NLP (Kalchbrenner, Grefenstette, & Blunsom, 2014), that automatically learns the features from the text, and minimizes the effort spent on feature extraction. Instead of producing a large vector representation for each sample, i.e., tens of thousands of dimensions, this model employs three much smaller word embedding vectors with just a few hundred dimensions. Given a sentence with marked entities, each word in the sentence is represented by a low-dimension vector concatenated from (1)the word embedding, (2) the relative position embedding, and (3) the entity type embedding. The vectors of words then form a matrix working as the representation of the

sentence. The matrix is then fed to multiple stacks of a convolutional layer, a max-pooling layer, and a fully connected layer. The model is trained using the gradient descent algorithm with cross-entropy loss. Some regularization techniques are applied to improve the model, such as mini-batch training, adaptive learning rate optimizer, and weight normalization.

Many efforts have introduced different pooling techniques to extract meaningful information for event extract from what is provided in the sentence. Y. Chen, Xu, Liu, Zeng, and Zhao (2015) improved the CNN model by using multi-pooling (DMCNN) instead of vanilla max-pooling. In this model, the sentence is split into multiple parts by either the examining event trigger or the given entity markers. The pooling layer is applied separately on each part of the sentence. Z. Zhang, Xu, and Chen (2016) proposed skip-window convolution neural networks (S-CNNs) to extract global structured features. The model effectively captures the global dependencies of every token in the sentence. L. Li, Liu, and Qin (2018) proposed a parallel multi-pooling convolutional neural network (PMCNN) that applies not only multiple pooling for the examining event trigger and entities but also to every other trigger and argument that appear in the sentence. This helps to capture the compositional semantic features of the sentence.

Kodelja, Besançon, and Ferret (2019) integrated the global representation of contexts beyond the sentence level into the convolutional neural network. To generate the global representation in connection with the target event detection task, they label the whole given document using a bootstrapping model. The bootstrapping model is based on the usual CNN model. The predictions for every token are aggregated to generate the global representation.

Even though CNN, together with the distributed word representations, can automatically capture local features, EE models based on CNN are not successful at capturing long-range dependency between words. The reason is that CNN can only model the short-range dependencies within the window of its kernel. Moreover, a large amount of information is lost because of the pooling operations (e.g., max pooling). As such, a more sophisticated neural network design is needed to model the long-range dependency between words in long sentences and documents without sacrificing information.

1.4.2.3 Recurrent Neural Networks.

T. H. Nguyen, Cho, and Grishman (2016) employed Gated Recurrent Unit (GRU) (Cho, van Merriënboer, Bahdanau, & Bengio, 2014), an RNN-based architecture, to better model the relation between words in a sentence. The model produces a rich representation based on the context captured in the sentence for the prediction of event triggers and event arguments. The model includes two recurrent neural networks, one for the forward direction and one for the backward direction.

Sentence embedding: Similar to CNN model, each word w_i of the sentence is transformed into a fixed-size real-value vector x_i. The feature vector is a concatenation of the word embedding vector of the current word, the embedding vector for the entity type of the current word, and the one-hot vector whose dimensions correspond to the possible relations between words in the dependency trees.

RNN encoding: The model employs two recurrent networks, forward and backward, denoted as $\overrightarrow{RNN}$ and $\overleftarrow{RNN}$ to encode the sentence word-by-word:

$$(a_1, \cdots, a_N) = \overrightarrow{RNN}(x_1, \cdots, x_N)$$

$$(a'_1, \cdots, a'_N) = \overleftarrow{RNN}(x_1, \cdots, x_N)$$

Finally, the representation h_i for each word is the concatenation of the corresponding forward and backward vectors $h_i = [a_i, a'_i]$.

Prediction: To jointly predict the event triggers and arguments, a binary vector for trigger and two binary matrices are introduced for event arguments. These vectors and matrices are initialized to zero. For each iteration, according to each word w_i, the prediction is made in a 3-step process: trigger prediction for w_i, argument role prediction for all the entity mentions given in the sentence, and finally, compute the vector and matrices of the current step using the memory and the output of the previous step.

Similarly, Ghaeini, Fern, Huang, and Tadepalli (2016) and Y. Chen, Liu, He, Liu, and Zhao (2016) employed Long Short-Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997), anther architecture based on RNN. LSTM is much more complex than the original RNN architecture and the GRU architecture. LSTM can capture the semantics of words with consideration of the context given by the context words automatically. Y. Chen et al. (2016) further proposed Dynamic Multi-Pooling similar to the DMCNN (Y. Chen et al., 2015) to extract event and argument separately. Furthermore, the model proposed a tensor layer to model the interaction between candidate arguments.

Even though the vanilla LSTM (or sequential/linear LSTM) can capture a longer dependency than CNN, in many cases, the event trigger and its arguments are distant. As such, the LSTM model can not capture the dependency between them. However, the distance between those words is much shorter in a dependency tree. Using a dependency tree to represent the relationship between words in the sentence can bring the trigger and entities close to each other. Some studies have implemented this structure in various ways. Sha, Qian, Chang, and Sui (2018)

proposed to enhance the bidirectional RNN with dependency bridges, which channel the syntactic information when modeling words in the sentence. They illustrate that simultaneously employing hierarchical tree structure and sequence structure in RNN improves the model's performance against the conventional sequential structure. D. Li, Huang, Ji, and Han (2019) introduced tree a knowledge base (KB)-driven tree-structured long short-term memory networks (Tree-LSTM) framework. This model incorporates two new features: dependency structures to capture broad contexts and entity properties (types and category descriptions) from external ontologies via entity linking.

1.4.3 Graph Convolutional Neural Networks.

The presented CNN-based and LSTM-based models for event detection have only considered the sequential representation of sentences. However, in these models, graph-based representation such as syntactic dependency tree (Nivre et al., 2016) has not been explored for event extraction, even though they provide an effective mechanism to link words to their informative context in the sentences directly.

For example, Figure 1 presents the dependency tree of the sentence "*This LPA-induced rapid phosphorylation of radixin was significantly suppressed in the presence of C3 toxin, a potent inhibitor of Rho*". In this sentence, there is a event trigger "*suppressed*" with its argument "*C3 toxin*". In the sequential representation, these words are 5-step apart, whereas in the dependency tree, they are 2-step apart. This example demonstrates the potential of the dependency tree in extracting event triggers and their arguments.

Many EE studies have widely used graph convolutional neural networks (GCN) (Kipf & Welling, 2017). It features two main ingredients: a convolutional

operation and a graph. The convolutional operation works similarly in both CNNs and GCNs. It learns the features by integrating the features of the neighboring nodes. In GCNs, the neighborhoods are the adjacent nodes on the graph, whereas, in CNNs, the neighborhoods are surrounding words in linear form.

Formally, let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph, and A be its adjacency matrix. The output of the $l+1$ convolutional layer on a graph $\mathcal{G}$ is computed based on the hidden states $H^l = \{h_i^l\}$ of the l-th layer as follows:

$$h_i^{l+1} = \sigma \sum_{(i,j)\in\mathcal{E}} \alpha_{ij}^l W^l h_j^l + b^l \tag{1.1}$$

Or in matrix form:

$$H^{l+1} = \sigma(\alpha^l W^l H^l A + b^l) \tag{1.2}$$

where W and b are learnable parameters and σ is a non-linear activation function; α_{ij} is the weight for the edge ij, in the simplest way, $\alpha_{ij} = 1$ for all edges.

GCN-ED (T. H. Nguyen & Grishman, 2018) and JMEE (X. Liu, Luo, & Huang, 2018) models are the first to use GCN for event detection. The graph used in the model is based on a transformation of the syntactic dependency tree. Let $\mathcal{G}_{\text{dep}} = (\mathcal{V}, \mathcal{E}_{\text{dep}})$ be an acyclic directed graph, representing the syntactic dependency tree of a given sentence. $\mathcal{V} = \{w_i | i \in [1, N]\}$ is the set of nodes; $\mathcal{E}_{\text{dep}} = \{(w_i, w_j) | i, j \in [1, N]\}$ is the set of edges. Each node of the graph represents a token in the given sentence, whereas each directed edge represents a syntactic arc in the dependency tree. The graph G used in GCN-ED and JMEE is derived with two main improvements:

– For each node w_i, a self-loop edge (w_i, w_i) is added to the set of edges so that the representation of the node is computed of the representation of itself.

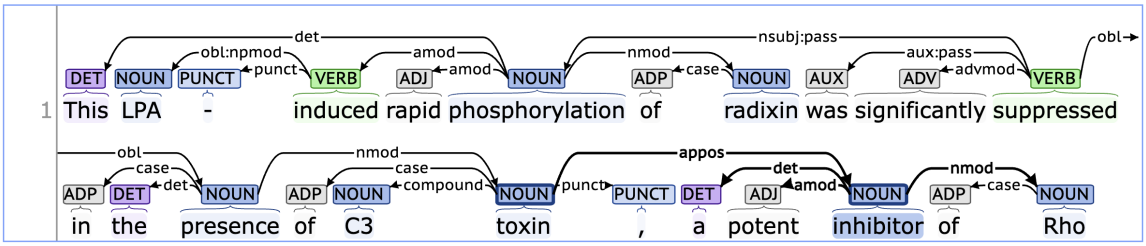

Figure 1. Dependency tree for sentence "*This LPA-induced rapid phosphorylation of radixin was significantly suppressed in the presence of C3 toxin, a potent inhibitor of Rho*", parsed by Trankit toolkit.

– For each edge (w_i, w_j), a reverse edge (w_j, w_i) of the same dependency type is added to the set of edges of the graph.

Mathematically, a new set of edge $\mathcal{E}$ is created as follows:

$$\mathcal{E} = \mathcal{E}_{\text{dep}} \cup \{(w_i, w_i) | w_i \in \mathcal{V}\}$$
$$\cup \{(w_j, w_i) | (w_i, w_j) \in \mathcal{E}_{\text{dep}}\}$$

Once the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is created, the convolutional operation, as shown in Equation 1.1 is applied multiple times on the input word embedding. Due to the small scale of the ED dataset, instead of using different sets of weights and biases for each dependency relation type, T. H. Nguyen and Grishman (2018) used only three sets of weights and biases for three types of dependency edges based on their origin: the original edges from $\mathcal{E}_{\text{dep}}$, the self-loop edges, and the inverse edges.

In the dependency graph, some neighbors of a node could be more important for event detection than others. Inspired by this, T. H. Nguyen and Grishman (2018) and X. Liu et al. (2018) also introduced neighbor weighting (Marcheggiani & Titov, 2017), in which neighbors are weighted differently depending on the level of importance. The weight α in Equation 1.1 is computed as follow:

$$\alpha_{ij}^l = \sigma(h_j^l W_{\text{type(i,j)}}^l) + b^l)$$

where h_j^l is the representation of the j-th words at the l-th layer. $W^l_{\text{type(i,j)}}$ and b^l are weight and bias terms, and σ is a non-linear activation function.

However, the above dependency-tree-based methods explicitly use only first-order syntactic edges, although they may also implicitly capture high-order syntactic relations by stacking more GCN layers. As the number of GCN layers increases, the representations of neighboring words in the dependency tree will get more and more similar since they all are calculated via those of their neighbors in the dependency tree, which damages the diversity of the representations of neighboring words. As such, Yan, Jin, Meng, Guo, and Cheng (2019) introduced Multi-Order Graph Attention Network for Event Detection (MOGANED). In this model, the hidden vectors are computed based on the representations of not only the first-order neighbors but also higher-order neighbors in the syntactic dependency graph. To do that, they used Graph Attention network (GAT) (Veličković et al., 2018) and an attention aggregation mechanism to merge its multi-order representations.

In a multi-layer GCN model, each layer has its scope of neighboring. For example, the representation of a node in the first layer is computed from the representations of its first-order neighbors only, whereas one in the second layer is computed from the representations of both first-order and second-order neighbors. As such, V. D. Lai, Nguyen, and Nguyen (2020a) proposed GatedGCN with an enhancement to the graph convolutional neural network with layer diversity using a gating mechanism. The mechanism helps the model to distinguish the information derived from different sources, e.g., first-order neighbors and second-order neighbors. The authors also introduced importance score consistency between model-predicted importance scores and graph-based importance scores. The graph-based importance

scores are computed based on the distances between nodes in the dependency graph.

The above GCN-based models usually ignore dependency label information, which conveys rich and useful linguistic knowledge for ED. Edge-Enhanced Graph Convolution Network (EE-GCN), on the other hand, simultaneously exploited syntactic structure and typed dependency label information (Cui et al., 2020). The model introduces a mechanism to dynamically update the representation of node-embedding and edge-embedding according to the context presented in the neighboring nodes. Similarly, Dutta et al. (2021) presented the GTN-ED model that enhanced prior GCN-based models using dependency edge information. In particular, the model learns a soft selection of edge types and composite relations (e.g., multi-hop connections, called meta-paths) among the words, thus producing heterogeneous adjacency matrices.

1.4.4 Knowledge Base.

As mentioned before, event extraction extract events from the text that involves some named entities such as participants, time, and location. In some domains, such as the biomedical domain, it requires a broader knowledge acquisition and a deeper understanding of the complex context to perform the event extraction task. Fortunately, a large number of those entities and events have been recorded in existing knowledge bases. Hence, these knowledge bases may provide the model with a concrete background of the domain terminologies as well as their relationship. This section presents some methods to exploit external knowledge to enhance event extraction models.

D. Li et al. (2019) proposed a model to construct knowledge base concept embedding to enrich the text representation for the biomedical domain. In

particular, to better capture domain-specific knowledge, the model leverages the external knowledge bases (KBs) to acquire properties of all the biomedical entities. Gene Ontology is used as their external knowledge base because it provides detailed gene information, such as gene functions and relations between them as well as gene product information, e.g., related attributes, entity names, and types. Two types of information are extracted from the KB to enrich the feature of the model: (1) entity type and (2) gene function description. First, the entity type for each entity is queried, then it is injected into the model similar to (T. H. Nguyen & Grishman, 2015). Second, the gene function definition, which is usually a long phrase, is passed through a language model to obtain the embedding. Finally, the embedding is concatenated to the input representation of the LSTM model.

K.-H. Huang, Yang, and Peng (2020), on the other hand, argues that the word embedding does not provide adequate clues for event extraction in extreme cases such as non-indicative trigger words and nested structures. For example, in the biomedical domain, many entities have hierarchical relations that might help to provide domain knowledge to the model. In particular, the Unified Medical Language System (UMLS) is the knowledge base that is used in this study. UMLS provides a large set of medical concepts, their pair-wise relations, and relation types. To incorporate the knowledge, words in the sentence are mapped to the set of concepts, if applicable. Then they are connected using the relations provided by the KB to form a semantic graph. This graph is then used in their graph neural network.

1.4.5 Data Generation.

As shown in Section 1.3, most of the datasets for Event Extraction were created based on human annotation, which is very laborious. As such, these

datasets are limited in size, as shown in Table 4. Moreover, these datasets are usually extremely imbalanced. These issues might hinder the learning process of the deep neural network. Many methods of data generation have been introduced to enlarge the EE datasets, which results in significant improvement in the performance of the EE model.

External knowledge bases such as Freebase, Wikipedia, and FrameNet are commonly used in event generation. S. Liu, Chen, He, Liu, and Zhao (2016) trained an ED model on the ACE dataset to predict the event label on FrameNet text to produce a semi-supervised dataset. The generated data was then further filtered using a set of global constraints based on the original annotated frame from FrameNet. L. Huang et al. (2016), on the other hand, employs a word-sense disambiguation model to predict the word-sense label for unlabeled text. Words that belong to a subset of verb and noun senses are considered as trigger words. To identify the event arguments for the triggers, the text is parsed into an AMR graph that provides arguments for trigger candidates. The argument role is manually mapped from AMR argument types. Y. Chen et al. (2017); Zeng et al. (2018) proposed to automatically label training data for event extraction based on distant supervision via Freebase, Wikipedia, and FrameNet data. The Freebase provides a set of key arguments for each event type. After that, candidate sentences are searched among Wikipedia text for the appearances of key arguments. Given the sentence, the trigger word is identified by a strong heuristic rule.

Ferguson, Lockard, Weld, and Hajishirzi (2018) proposed to use bootstrapping for event extraction. The core idea is based on the occurrence of multiple mentions of the same event instances across newswire articles from multiple sources. Hence, if an ED model detects some event mentions at high

confidence from a cluster, the model can then acquire diverse training examples by adding the other mentions from that cluster. The authors trained an ED model based on limited available training data and then used that model for data labeling on unlabeled newswire text.

S. Yang, Feng, Qiao, Kan, and Li (2019) explored the method that uses a generative model to generate more data. They generated data from the golden ACE dataset in three steps. First, the arguments in a sentence are replaced with highly similar arguments found in the golden data to create a noisy sentence. Second, a language model is used to regenerate the sentence from the noisy generated sentence to create a new smoother sentence to avoid overfitting. Finally, the candidate sentences are ranked using a perplexity score to find the best-generated sentence.

Tong et al. (2020) argued that open-domain trigger knowledge could alleviate the lack of data and training data imbalance in the existing EE dataset. The authors proposed a novel Enrichment Knowledge Distillation (EKD) model that can generate noisy ED data from unlabeled text. Unlike the prior methods that employed rules or constraints to filter noisy data, their model used the teacher-student model to automatically distill the training data.

1.4.6 Document-level Modeling.

The methods for event extraction mentioned so far have not gone beyond the sentence level. Unfortunately, this is a systematic problem as, in reality, events and their associated arguments can be mentioned across multiple sentences in a document (H. Yang, Chen, Liu, Xiao, & Zhao, 2018). Hence, such sentence-level event extraction methods struggle to handle documents in which events and their arguments scatter across multiple sentences. The document-level event extraction

(DEE) paradigm has been investigated to address the problem of sentence-level event extraction. Many researchers have proposed methods to model document-level relations such as entity interactions, sentence interactions (Y. Huang & Jia, 2021; Xu et al., 2021), reconstruct document-level structure (K.-H. Huang & Peng, 2021), and model long-range dependencies while encoding a lengthy document (Du & Cardie, 2020).

Initial studies for DEE did not consider modeling the document-level relation properly. H. Yang et al. (2018) was the first attempt to explore the DEE problem on a Chinese Financial Document corpus (ChiFinAnn) by generating weakly-supervised EE data using distant supervision. Their model performs DEE in two stages. First, a sequence tagging model extracts events at the sentence level in every document sentence. Second, key events are detected among extracted events, and arguments are heuristically collected from all over the document. Zheng, Cao, Xu, and Bian (2019), on the other hand, proposed an end-to-end model named Doc2EDAG. The model encodes documents using a transformer-based encoder. Instead of filling the argument table, they created an entity-based directed acyclic graph to find the argument effectively through path expansion. Du and Cardie (2020) transforms the role filler extraction into an end-to-end neural sequence learning task. They proposed a multi-granularity reader to efficiently collect information at different levels of granularity, such as sentence and paragraph levels. Therefore, it mitigates the effect of long dependencies of scattering argument in DEE.

Some studies have attempted to exploit the relationship between entities, event mentions, and sentences of the document. Y. Huang and Jia (2021) modeled the interactions between entities and sentences within long documents. In

particular, instead of constructing an isolated graph for each sentence, this work constructs a unified unweighted graph for the whole document by exploiting the relationship between sentences. Furthermore, they proposed the sentence community consisting of sentences related to the same event's arguments. The model detects multiple event mentions by detecting those sentence communities. To encourage the interaction between entities, Xu et al. (2021) proposed a Heterogeneous Graph-based Interaction Model with a Tracker (GIT) to model the global interaction between entities in a document. The graph leverages multiple document-level relations, including sentence-sentence edges, sentence-mention edges, intra mention-mention edges, and inter mention-mention edges. K.-H. Huang and Peng (2021) introduced an end-to-end model featuring a structured prediction algorithm, Deep Value Networks, to efficiently model cross-event dependencies for document-level event extraction. The model jointly learns entity recognition, event co-reference, and event extraction tasks, resulting in a richer representation and a more robust model.

1.4.7 Joint Modeling.

The above works have executed the four subtasks of event extraction in a pipeline where the model uses the prediction of other models to perform its task. Consequently, the errors of the upstream subtasks are propagated through the downstream subtasks in the pipeline, ruining their performances. Additionally, the knowledge learned from the downstream subtasks can not influence the prediction decision of the upstream subtasks. Thus, the dependence on the tasks can not be exploited thoroughly. To address the issues of the pipeline model, joint modeling of multiple event extraction subtasks is an alternative to take advantage of the interactions between the EE subtasks. The interactions between subtasks are

bidirectional. Therefore, useful information can be carried across the subtasks to alleviate error propagation.

Joint modeling can be used to train a diverse set of subtasks. For example, H. Lee, Recasens, Chang, Surdeanu, and Jurafsky (2012) trained a joint model for event co-reference resolution and entity co-reference resolution, while R. Han, Ning, and Peng (2019) proposed a joint model for event detection and event temporal relation extraction. In the early day, modeling event detection and argument role extraction together are very popular (Q. Li, Ji, & Huang, 2013; T. H. Nguyen, Cho, & Grishman, 2016; Venugopal, Chen, Gogate, & Ng, 2014). Recent joint modeling systems have trained models with up to 4 subtasks (i.e. event detection, entity extraction, event argument extraction, and entity linking) (Lin, Ji, Huang, & Wu, 2020; M. V. Nguyen, Lai, & Nguyen, 2021; M. V. Nguyen, Min, Dernoncourt, & Nguyen, 2022; Z. Zhang & Ji, 2021). Table 5 presents a summary of the subtasks that were used for joint modeling for EE.

Early joint models were simultaneously trained to extract the trigger mention and the argument role (Q. Li et al., 2013), Q. Li et al. (2013) formulated a two-task problem as a structural learning problem. They incorporated both global features and local features into a perceptron model. The trigger mention and arguments are decoded simultaneously using a beam search decoder. Later models that are based on a neural network share a sentence encoder for all the subtasks (R. Han et al., 2019; T. H. Nguyen, Cho, & Grishman, 2016; Wadden, Wennberg, Luan, & Hajishirzi, 2019) so that the training signals of different subtasks can impact the representation induced by the sentence encoder.

Besides the shared encoders, recent models use various techniques to encourage interactions between subtasks. T. H. Nguyen, Cho, and Grishman

(2016) employed a memory matrix to memorize the dependencies between event and argument labels. These memories are then used as a new feature in the trigger and argument prediction. They employed three types of dependencies: (i) trigger subtype dependency, (ii) argument role dependency, and (iii) trigger-argument role dependency. These terminologies were later generalized as intra/inter-subtask dependencies (Lin et al., 2020; M. V. Nguyen, Lai, & Nguyen, 2021; M. V. Nguyen et al., 2022).

Luan et al. (2019) proposed the DyGIE model that employed an interactive graph-based propagation between events and entities nodes based on entity co-references and entity relations. In particular, in DyGIE model (Luan et al., 2019), the input sentences are encoded using a BiLSTM model, then, a contextualized representation is computed for each possible text span. They employed a dynamic span graph whose nodes are selectively chosen from the span pool. At each training step, the model updates the set of graph nodes. It also constructs the edge weights for the newly created graph. Then, the representations of spans are updated based on neighboring entities and connected relations. Finally, the predictions of entities, events, and their relations are based on the latest representations. Wadden et al. (2019) further improved the model with contextualized embeddings BERT while maintaining the core architecture of DyGIE. Even though these models have introduced task knowledge interaction through graph propagation, their top task prediction layers still make predictions independently. In other words, the final prediction decision is still made locally.

To address the DyGIE/DyGIE++ issue, OneIE model (Lin et al., 2020) proposed to enforce global constraints to the final predictions. They employed a beam search decoder at the final prediction layer to globally constrain the

predictions of the subtasks. Similar to JREE model (T. H. Nguyen, Cho, & Grishman, 2016), they considered both cross-subtask interactions and cross-instance interactions. To do that, they designed a set of global feature templates to capture both types of interactions. Given all the templates, the model tries to fill all possible features and learns the weights. To make the final prediction, a trivial solution is an exhaustive search during the inference. However, the search space grows exponentially, leading to an infeasible problem. They proposed a graph-based beam search algorithm to find the optimal graph. In each step, the beam grows with either a new node (i.e., a trigger or an entity) or a new edge (i.e., an argument role or an entity relation).

In the above neural-based models, the predictive representation of the candidates is computed independently using contextualized embedding. Consequently, the predictive representation has not considered the representations of the other related candidates. FourIE model (M. V. Nguyen, Lai, & Nguyen, 2021) features a graph structure to encourage interactions between related instances of a multi-task EE problem. M. V. Nguyen, Lai, and Nguyen (2021) further argued that the global feature constraint in OneIE (Lin et al., 2020) is suboptimal because it is manually created. They instead introduced an additional graph-based neural network to score the candidate graphs. To train this scoring network, they employ Gumbel-Softmax distribution (Jang, Gu, & Poole, 2017) to allow gradient updates through the discrete selection process. However, due to the heuristical design of the dependency graph, the model may fail to explore other possible interactions between the instances. As such, M. V. Nguyen et al. (2022) explicitly model the dependencies between tasks by modeling each task instance as a node in the fully connected dependency graph. The weight for each edge is learnable, allowing a soft

interaction between instances instead of hard interactions in prior works (Lin et al., 2020; M. V. Nguyen, Lai, & Nguyen, 2021; Z. Zhang & Ji, 2021)

Recently, joint modeling for event extraction was formulated as a text generation task using pre-trained generative language models such as BART (Lewis et al., 2020), and T5 (Raffel et al., 2020). In these models (Hsu et al., 2022; Y. Lu et al., 2021), the event mentions, entity mentions, as well as their labels and relations are generated by an attention-based autoregressive decoder. The task dependencies are encoded through the attention mechanism of the transformer-based decoder. This allows the model to learn the dependencies between tasks and task instances flexibly. However, to train the model, they have to assume an order of tasks and task instances that are being decoded. As a result, the model suffers from the same problem that arose in pipeline models.

1.5 Low-resource Event Extraction

State-of-the-art event extraction approaches, which follow the traditional supervised learning paradigm, require great human efforts to create high-quality annotation guidelines and annotate the data for a new event type. For each event type, language experts need to write annotation guidelines that describe the class of event and distinguish it from the other types. Then annotators are trained to label event triggers in the text to produce a large dataset. Finally, a supervised-learning-based classifier is trained on the obtained event triggers to label the target event. This labor-exhaustive process might limit the applications of event extraction in real-life scenarios. As such, approaches that require less data creation are becoming more and more attractive thanks to their fast deployment and low-cost solution. However, this line of research faces a challenging wall due to their limited access to labeled data. This section presents recent studies on low-resource event extraction

in various learning paradigms and domains. The rest of the section is organized as follow: Section 1.5.1 highlights some methods of zero-shot learning; section 1.5.2 presents a new clusters of recent studies in few-shot learning. Finally, methods for cross-lingual event extraction is presented in section 1.5.3.

1.5.1 Zero-shot Learning.

Zero-shot learning (ZSL) is a type of transfer learning in which a model performs a task without any training samples. Toward this end, transfer learning uses a pre-existing classifier to build a universal concept space for both seen and unseen samples. Existing methods for event extraction exploits latent-variable space in CRF model (W. Lu & Roth, 2012), rich structural features such as dependency tree and AMR graph (L. Huang et al., 2018), ontology mapping (H. Zhang, Wang, & Roth, 2021), and casting the problem into a question-answering problem (J. Liu, Chen, Liu, Bi, & Liu, 2020; Lyu, Zhang, Sulem, & Roth, 2021).

The early study by W. Lu and Roth (2012) showed the first attempt to solve the event extraction problem under zero-shot learning. They proposed to model the problem using latent variable semi-Markov conditional random fields. The model jointly extracts event mentions and event arguments given event templates, coarse event/entity mentions, and their types. They used a framework called structured Preference Modeling (PM). This framework allows arbitrary preferences associated with specific structures during the training process.

Inspired by the shared structure between events, L. Huang et al. (2018) introduced a transfer learning method that matches the structural similarity of the event in the text. They proposed a transferable architecture of structural and compositional neural networks to jointly produce to represent event mentions,

their types, and their arguments in a shared latent space. This framework allows for predicting the semantically closest event types for each event mention. Hence, this framework can be applied to unseen event types by exploiting the limited manual annotations. In particular, event and argument candidates are detected by exploiting the AMR graph of the sentence. After this, a CNN is used to encode all the triplets representing AMR edges, e.g. (dispatch-01, :ARG0, China). For each new event type, the same CNN model encodes the relations between event type, argument role, and entity type, e.g. (Transport Person, Destination), resulting in a representation vector for the new event ontology. The model chooses the closest event type based on the similarity score between the trigger's encoded vector and all available event ontology vectors to predict the event type for a candidate event trigger.

H. Zhang et al. (2021) proposed a zero-shot event extraction method that (1) extracts the event mentions using existing tools, then, and (2) maps these events to the targeted event types with zero-shot learning. Specifically, an event-type representation is induced by a large pre-trained language model using the event definition for each event type. Similarly, event mentions and entity mentions are encoded into vectors using a pre-trained language model. Initial predictions are obtained by computing the cosine similarities between label and event representations. To train the model, an ILP solver is employed to regulate the predictions according to the given ontology of each event type. In detail, they used the following constraints: (1) one event type per event mention, (2) one argument role per argument, (3) different arguments must have different types, (4) predicted triggers and argument types must be in the ontology, and (5) entity type of the argument must match the requirement in the ontology.

Thanks to the rapid development of large generative language models, a language model can embed texts and answer human-language questions in a human-friendly way using its large deep knowledge obtained from massive training data. J. Liu et al. (2020) proposed a new learning setting of event extraction. They cast it as a machine reading comprehension problem (MRC). The modeling includes (1) an unsupervised question generation process, which can transfer event schema into a set of natural questions, and (2) a BERT-based question-answering process to generate the answers as EE results. This learning paradigm exploits the learned knowledge of the language model and strengthens EE's reasoning process by integrating sophisticated MRC models into the EE model. Moreover, it can alleviate the data scarcity issue by transferring the knowledge of MRC datasets to train EE models. Lyu et al. (2021), on the other hand, explore the Textual Entailment (TE) task and/or Question Answering (QA) task for zero-shot event extraction. Specifically, they cast the event trigger detection as a TE task, in which the TE model predicts the level of entailment of a hypothesis (e.g., *This is about a birth event* given a premise, i.e., the original text. Since an event may associate with multiple arguments, they cast the event argument extraction into a QA task. Given an input text and the extracted event trigger, the model is asked a set of questions based on the event type definition in the ontology, and retrieve the QA answers as predicted argument.

1.5.2 Few-shot Learning. There are several ways of modeling the event detection in a few-shot learning scheme (FSL-ED): (1) token classification FSL-ED and (2) sequence labeling FSL-ED.

Most of the studies following token classification setting (Bronstein, Dagan, Li, Ji, & Frank, 2015; Deng et al., 2020; V. D. Lai & Nguyen, 2019;

V. D. Lai, Nguyen, & Dernoncourt, 2020; Peng, Song, & Roth, 2016) are based on a prototypical network (Snell, Swersky, & Zemel, 2017), which employs a general-purpose event encoder for embed event candidates while the predictions are done using a non-parameterized metric-based classifier. Since the classifiers are non-parametric, these studies mainly explore the methods to improve the event encoder.

Bronstein et al. (2015) were among the first working in few-shot event detection. They proposed a different training/evaluation for event detection with minimal supervision. They proposed an alternative method, which uses the trigger terms included in the annotation guidelines as seeds for each event type. The model consists of an encoder and a classifier. The encoder embeds a trigger candidate into a fix-size embedding vector. The classifier is an event-independent similarity-based classifier. This work argues that they can eliminate the costly manual annotation for new event types. At the same time, the non-parametric classifier does not require a large amount to be trained, in fact, just a few event examples at the beginning. Peng et al. (2016) addressed the manual annotation by proposing an event detection and coreference system that requires minimal supervision, particularly a few training examples. Their approach was built on a key assumption: the semantics of two tasks (i) identifying events closely related to some event types and (ii) event coreference are similar. As such, reformulating the task into semantic similarity can help the model to be trained on a large available corpus of event coreference instead of annotating a large dataset for event detection. As a result, the required data for any new event type is as small as the number of samples in the annotation guidelines. To do that, they use a general purpose nominal and verbial semantic role labeling (SRL) representation to represent the structure of an event. The representation involves multiple semantic spaces,

including contextual, topical, and syntactic levels. Similarly, V. D. Lai and Nguyen (2019) proposed a novel formulation for event detection, namely learning from keywords (LFK) in which each type is described via a few event triggers. They are pre-selected from a pool of known events. In order to encode the sentence, the model contains a CNN-based encoder and a conditional feature-wise attention mechanism to selectively enhance informative features.

V. D. Lai, Nguyen, and Dernoncourt (2020), Deng et al. (2020) and V. Lai, Dernoncourt, and Nguyen (2021) employed the core architecture of the prototypical network while proposed an auxiliary training loss factors during the training process. V. D. Lai, Nguyen, and Dernoncourt (2020) enforce the distances between clusters of samples, namely intra-cluster loss and inter-cluster loss. The intra-cluster loss minimizes the distances between samples of the same class. In contrast, the inter-cluster loss maximizes the distances between the prototype of a class and the examples of the other classes. The model also introduces contextualized embedding, which leads to significant performance improvement over ANN or CNN-based encoders. Deng et al. (2020), on the other hand, proposed a Dynamic-Memory-Based Prototypical Network (DMB-PN). The model uses a Dynamic Memory Network(DMN) to learn better prototypes and produce better event mention encodings. The prototypes are not computed by averaging the supporting events just once, but they are induced from the supporting events multiple times through DMN's multihop mechanism. V. Lai et al. (2021) addressed the outlier and sampling bias in the training process of few-shot event detection. Particularly, in event detection, a null class is introduced to represent samples that are out of the interested classes. These may contain non-interested eventive samples as well as non-eventive samples. As such, this class may inject outlier examples into the

support set. As such, they proposed a novel model for the relation between two training tasks in an episodic training setting by allowing interactions between prototypes of two tasks. They also proposed prediction consistency between two tasks so that the trained model would be more resistant to outliers.

J. Chen, Lin, Han, and Sun (2021) addressed the trigger curse problem in FSL-ED. Particularly, both overfitting and underfitting trigger identification are harmful to the generalization ability or the detection performance of the model, respectively. They argue that the trigger is the confounder of the context and the result of an event. As such, previous models, which are trigger-centric, can easily overfit triggers. To alleviate the trigger overfitting, they proposed a method to intervene in the context by backdoor adjustment during training.

Recent work by Shen et al. (2021) tackles the low sample diversity in FSL-ED. Their model, Adaptive Knowledge-Enhanced Bayesian Meta-Learning (AKE-BML), introduces external event knowledge as a prior of the event type. First, they heuristically align the event types in the support set and FrameNet to do that. Then they encode the samples and the aligned examples in the same semantic space using a neural-based encoder. After that, they realign the knowledge representation by using a learnable offset, resulting in a prior knowledge distribution for event types. Then they can generate a posterior distribution for event types. Finally, to predict the label for a query instance, they use the posterior distribution for prototype representations to classify query instances into event types.

The second FSL-ED setting is based on sequence labeling. The few-shot sequence labeling setting, in general, has been widely studied in named entities recognition (Fritzler, Logacheva, & Kretov, 2019). Similarly, Cong et al. (2021) formulated the FSL-ED as a few-shot sequence labeling problem, which detects the

spans of the events and the label of the event at the same time. They argue that previous studies that solve this problem in the **identify-then-classify** manner suffer from error propagation due to ignoring the discrepancy of triggers between event types. They proposed a CRF-based model called Prototypical Amortized Conditional Random Field (PA-CRF). In order to model the CRF-based classifiers, it is important to approximate the transition and emission scores from just a few examples. Their model approximates the transition scores between labels based on the label prototypes. In the meantime, they introduced a Gaussian distribution into the transition scores to alleviate the uncertain estimation of the emission scorer.

1.5.3 Cross-lingual.

Early studies of cross-lingual event extraction (CLEE) relies on training a statistical model on parallel data for event extraction (Z. Chen & Ji, 2009; Hsi, Yang, Carbonell, & Xu, 2016; Piskorski, Belayeva, & Atkinson, 2011). Recent methods focus on transferring universal structures across languages (J. Liu, Chen, Liu, & Zhao, 2019; D. Lu et al., 2020; M. V. Nguyen & Nguyen, 2021; Subburathinam et al., 2019). There are a few other methods were also studied such as topic modeling (H. Li, Ji, Deng, & Han, 2011), multilingual embedding (M'hamdi, Freedman, & May, 2019), and annotation projection (F. Li, Huang, Xiong, & Zhang, 2016; Lou et al., 2022).

Cross-lingual event extraction depends on a parallel corpus for both training and evaluation. However, parallel corpora for this area are scarce. Most of the work in CLEE were done using ACE-2005 (LDC, 2005), TAC-KBP (Mitamura, Liu, & Hovy, 2015, 2017), and TempEval-2 (Verhagen, Saurí, Caselli, & Pustejovsky, 2010). These multilingual datasets cover several popular languages, such as English, Chinese, Arabic, and Spanish. Recently, datasets that cover less common languages,

e.g., Polish, Danish, Turkish, Hindi, Urdu, Korean and Japanese, were created for event detection (Veyseh, Nguyen, Dernoncourt, & Nguyen, 2022) and event relation extraction (V. D. Lai, Veyseh, Nguyen, Dernoncourt, & Nguyen, 2022).

Due to data scarcity in target languages, the model trained on limited data might not be able to predict a wide range of events. Therefore, generating more data from the existing corpus in the source language is a trivial method. F. Li et al. (2016) proposed a projection algorithm to mine shared hidden phrases and structures between two languages (i.e., English and Chinese). They project seed phrases back and forth multiple rounds between the two languages using parallel corpora to obtain a diverse set of closely related phrases. The captured phrases are then used to train an ED model. This method was shown to effectively improve the diversity of the recognized events. Lou et al. (2022) addressed the problem of noise appearing in the translated corpus. They proposed an annotation projection approach that combines the translation projection and the event argument extraction task training step to alleviate the additional noise through implicit annotation projection. First, they translate the source language corpus into the target language using a multilingual machine translation model. To reduce the noise of the translated data, instead of training the model directly from them, they use multilingual embedding to embed the source language data and the translated derivatives in the target language into the same vector space. Their representations are then aligned using optimal transport. They proposed two additional training signals that either reduce the alignment scores or the prediction based on the aligned representation. Phung, Minh Tran, Nguyen, and Nguyen (2021) explored the cross-lingual transfer learning for event coreference resolution task. They introduced the language adversarial neural network to help the model distinguish

texts from the source and target languages. This helps the model improve the generalization over languages for the task. Similar to (Lou et al., 2022), the work by Phung et al. (2021) introduced an alignment method based on multiple views of the text from the source and the target languages. They further introduced optimal transport to better select edge examples in the source and target languages to train the language discriminator.

Multilingual embedding plays an important role in transferring knowledge between languages. There have been many multilingual contextualized embedding built for a large number of languages such as FastText (Joulin, Bojanowski, Mikolov, Jégou, & Grave, 2018), MUSE (Lample, Conneau, Denoyer, & Ranzato, 2017), mBERT (Devlin et al., 2019), mBART (Y. Liu et al., 2020), XLM-RoBERTa (Conneau et al., 2020), and mT5/mT6 (Chi et al., 2021; Xue et al., 2021). (M'hamdi et al., 2019) compared FastText, MUSE and mBERT. The results show that multilingual embeddings help transfer knowledge from English data to other languages, i.e., Chinese and Arabic. The performance boost is significant when all multilingual are added to train the model. Various multilingual embeddings have been employed in cross-lingual event extraction thanks to their robustness and transferability. However, models trained on multilingual embedding still suffer from performance drop in zero-shot cross-lingual settings. It is even worse than monolingual embedding if the monolingual model is trained on a large enough target dataset and a good enough monolingual contextualized embedding (V. D. Lai et al., 2022).

Most of the recent methods for cross-lingual event extraction are done via transferring shared features between languages, such as syntactic structures (e.g., part-of-speech, dependency tree), semantic features (e.g., contextualized

embedding), and relation structures (e.g., entity relation). Subburathinam et al. (2019) addressed the suitability of transferring cross-lingual structures for the event and relation extraction tasks. They exploit relevant language-universal features for relation and events such as symbolic features (i.e., part-of-speech and dependency path) and distributional features (i.e., type representation and contextualized representation) to transfer those structures appearing in the source language corpus to the target language. Thanks to this similarity, they encode all the entity mentions, event triggers, and event context from both languages into a complex shared cross-lingual vector space using a graph convolutional neural network. Hence, once the model is trained in English, this shared structural knowledge will be transferred to the target languages, such as Russian. (J. Liu et al., 2019) addressed two issues in cross-lingual transfer learning: (i) how to build a lexical mapping between languages and (ii) how to manage the effect of the word-order differences between different languages. First, they employ a context-dependent translation method to construct the lexical mapping between languages by first retrieving k nearest neighbors in a shared vector space, then reranking the candidates using a context-aware selective attention mechanism. To encode sentences with language-dependent word order, a GCN model is employed to encode the sentence. To enrich the features for the cross-lingual event argument extraction model, M. V. Nguyen and Nguyen (2021) employ three types of connection to build a feature-expanded graph. The core of the graph is derived from the dependency graph used in many other studies to capture syntactic features. They introduced two additional connections to capture semantic similarity and the universal dependency relations of the word pairs. Based on the assumption that most concepts are universal across languages, similarities between words and

representing concepts are also universal. They employ a multilingual contextualized embedding to obtain the word representation, and then compute a similarity score between words in a sentence. Secondly, they argue that the relation types play an important role in the connection's strength. Therefore, another connection set of weights is computed based on the dependency relation type between two connected words. Finally, the additional edge weights are added to the graph, scaling to the extent of the similarity score of the relation.

1.6 Conclusion

This chapter first states the topics and targets of this dissertation. After that, we present a comprehensive literature review of the existing work in Information Extraction ranging from early work with feature engineering, the use of deep neural network architecture, and recent advances in graph convolutional neural networks. The review spends a substantial effort in studies for low-resource event extraction and cross-lingual event extraction.

In the next chapter, since the graph convolutional neural network is widely used in information extraction research, we study a method to improve the performance of this model for EE.

Dataset	Topic	#Classes	#Samples	#Languages	Tasks
Event Extraction					
ACE-05	News	33	4,907	3	Trig, Arg, Ent, Rel, EntCoref
TAC-KBP	News	38	11,975	3	Trig, Arg, Ent, Rel, EntCoref
TimeBank	Newswire	8	7,935	1	Trig, Temporal
GENIA	Biomedical	36	36,114	1	Trig, Arg, Ent, Rel
CASIE	Cyber security	5	8,470	1	Trig
CyberED	Cyber security	30	8,014	1	Trig
Litbank	Literature	1	7,849	1	Trig, Ent, EntCoref
RAMS	News	139	9,124	1	Trig, Arg, Ent
BRAD	Black rebellion	12	4,259	1	Trig, Arg, Ent, Rel
SuicideED	Mental health	7	36,978	1	Trig, Arg, Ent, Rel
MAVEN	General	168	111,611	1	Trig
FedSemcor	General	449	34,666	1	Trig
MINION	Wikipedia	33	50,934	10	Trig
CLIP-Event	News	33	105,331	1	Trig, Arg, Ent
MEE	Wikipedia	16	50,011	8	Trig, Arg, Ent
Event Relation					
Causal-TimeBank	Newswire	-	318	1	Causal
RED		-	6,085	1	Causal, Temporal, Hierarchy
Because-2.0		-	1,803	1	Causal
CaTeRS		-	488	1	Causal
HiEve	News stories	-	2,257	1	Hierarchy, Coreference
TempEval	News	-		1	Temporal
EventStoryLine	Calamity events	-	8,201	1	Causal, Temporal
MATRES		-		1	Temporal
MECI	Wikipedia	-	11,055	5	Causal
mSubEvent	Wikipedia	-	3,944	5	Hierarchy
MAVEN-ERE	News	-	1,290,050	1	Causal, Temporal, Hierarchy, Coreference

Table 4. Statistics of existing event extraction datasets. Event-related tasks: Trigger Identification & Classification (Trig), Event Argument Extraction (Arg), Event Temporal (Temporal), Event Causality (Causal), Event Coreference (Coreference), Event Hierarchy (Hierarchy). Entity-related tasks: Entity Mention (Ent), Entity Linking (Rel), Entity Coreference (EntCoref).

Acronym	System	Event	Entity	Argument	Relation	EventCoref	EntityCoref	EventTemp
Lee's Joint	H. Lee et al. (2012)					✓	✓	
Li's Joint	Q. Li et al. (2013)	✓		✓				
MLN+SVM	Venugopal et al. (2014)	✓		✓				
Araki's Joint	Araki and Mitamura (2015)	✓				✓		
JRNN	T. H. Nguyen, Cho, and Grishman (2016)	✓	✓	✓				
Structure Joint	R. Han et al. (2019)	✓						✓
DyGIE	Luan et al. (2019)	✓	✓		✓			
DyGIE++	Wadden et al. (2019)	✓	✓		✓			
HPNet	P. Huang, Zhao, Takanobu, Tan, and Xiao (2020)	✓		✓				
OneIE	Lin et al. (2020)	✓	✓	✓	✓			
NGS	X. Wang, Jia, et al. (2020)	✓		✓				
Text2Event	Y. Lu et al. (2021)	✓		✓				
AMRIE	Z. Zhang and Ji (2021)	✓	✓	✓	✓			
FourIE	M. V. Nguyen, Lai, and Nguyen (2021)	✓	✓	✓	✓			
DEGREE	Hsu et al. (2022)	✓		✓				
GraphIE	M. V. Nguyen et al. (2022)	✓	✓	✓	✓			

Table 5. Subtasks for joint modeling in event extraction.

Model Acronym	System	Trigger		Argument	
		ID	C	ID	C
Feature engineering					
Ahn et al.	Ahn (2006)	62.6	60.1	82.4	57.3
Cross-document	Ji and Grishman (2008)	-	67.3	46.2	42.6
Cross-event	Liao and Grishman (2010)	-	68.8	50.3	44.6
Cross-entity	Hong et al. (2011)	-	68.3	53.1	48.3
Structure-prediction	Q. Li et al. (2013)	70.4	67.5	56.8	52.7
CNN					
CNN	T. H. Nguyen and Grishman (2015)		69.0	-	-
DMCNN	Y. Chen et al. (2015)	73.5	69.1	59.1	53.5
DMCNN+DS	Y. Chen et al. (2017)	74.3	70.5	63.3	55.7
RNN					
JRNN	T. H. Nguyen, Cho, and Grishman (2016)	71.9	69.3	62.8	55.4
FBRNN	Ghaeini et al. (2016)	-	67.4	-	-
BDLSTM-TNNs	Y. Chen et al. (2016)	72.2	68.9	60.0	54.1
DLRNN	Duan, He, and Zhao (2017)	-	70.5	-	-
dbRNN	Sha et al. (2018)	-	71.9	67.7	58.7
GCN					
GCN-ED	T. H. Nguyen and Grishman (2018)	-	73.1	-	-
JMEE	X. Liu et al. (2018)	75.9	73.7	68.4	60.3
MOGANED	Yan et al. (2019)	-	75.7	-	-
MOGANED+GTN	Dutta et al. (2021)	-	76.8	-	-
GatedGCN	V. D. Lai, Nguyen, and Nguyen (2020a)	-	77.6	-	-
Data Generation & Augmentation					
ANN-FN	S. Liu et al. (2016)	-	70.7	-	-
Liberal	L. Huang et al. (2016)	-	61.8	-	44.8
Chen's Generation	Y. Chen et al. (2017)	74.3	70.5	63.3	55.7
BLSTM-CRF-ILP$_{multi}$	Zeng et al. (2018)	-	82.5	-	37.9
EKD	Tong et al. (2020)	-	78.6	-	-
GPTEDOT	Veyseh, Lai, Dernoncourt, and Nguyen (2021)	-	79.2	-	-
Document-level Modeling					
HBTNGMA	Y. Chen, Yang, Liu, Zhao, and Jia (2018)	-	73.3	-	-
DEEB-RNN	Zhao, Jin, Wang, and Cheng (2018)	-	74.9	-	-
ED3C	Veyseh, Nguyen, Ngo, Min, and Nguyen (2021)	-	79.1	-	-
Joint Modeling					
DyGIE++	Wadden et al. (2019)	76.5	73.6	55.4	52.5
HPNet	P. Huang et al. (2020)	79.2	77.8	60.9	56.8
OneIE	Lin et al. (2020)	-	72.8	-	56.3
NGS	X. Wang, Jia, et al. (2020)	-	74.6	-	59.5
Text2event	Y. Lu et al. (2021)	-	71.8	-	54.4
AMRIE	Z. Zhang and Ji (2021)	-	72.8	-	57.7
FourIE	M. V. Nguyen, Lai, and Nguyen (2021)	-	73.3	-	58.3
DEGREE	Hsu et al. (2022)	-	71.7	-	58.0
GraphIE	M. V. Nguyen et al. (2022)	-	74.8	-	60.2

Table 6. Summary of the performance of the EE models on the ACE-05 dataset for identification (ID) and classification (C) tasks.

CHAPTER II

GATE DIVERSITY AND SYNTACTIC IMPORTANCE SCORES FOR GRAPH CONVOLUTION NEURAL NETWORKS

This chapter contains materials from the published paper "*Lai, Viet Dac, Tuan Ngo Nguyen, and Thien Huu Nguyen.* **Event Detection: Gate Diversity and Syntactic Importance Scores for Graph Convolution Neural Networks.** *In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 5405-5411. 2020.*".

As the first author of this paper, Viet was responsible for the development, evaluation, and writing. Tuan and Thien provided meaningful discussion and analysis. Thien has put on editorial writing for the paper submission. The paper was revised to comply with the dissertation format and purposes.

After the literature review, this chapter presents the first contribution to representation learning of the models designed for Event Detection. In particular, we focus on a class of models based on graph convolutional neural networks that have been shown to effectively capture informative information for ED. However, the computation of the hidden vectors in such graph-based models is agnostic to the trigger candidate words, potentially leaving irrelevant information for the trigger candidate for event prediction. In addition, the current models for ED fail to exploit the overall contextual importance scores of the words, which can be obtained via the dependency tree, to boost the performance. In this study, we propose a novel gating mechanism to filter noisy information in the hidden vectors of the GCN models for ED based on the information from the trigger candidate. We also introduce novel mechanisms to achieve the contextual diversity for the gates and the importance score consistency for the graphs and models in ED. The

experiments show that the proposed model achieves state-of-the-art performance on two ED datasets.

2.1 Introduction

Event Detection (ED) is an important task in Information Extraction of Natural Language Processing. The main goal of this task is to identify event instances presented in text. Each event mention is associated with a word or a phrase, called an event trigger, which clearly expresses the event (Walker, Strassel, Medero, & Maeda, 2006). The event detection task, precisely speaking, seeks to identify the event triggers and classify them into some types of interest. For instance, consider the following sentences:

(1) *They'll be **fired** on at the crossing.*

(2) *She is on her way to get **fired**.*

An ideal ED system should be able to recognize the two words "*fired*" in the sentences as the triggers of the event types "Attack" (for the first sentence) and "End-Position" (for the second sentence).

The dominant approaches for ED involve deep neural networks to learn effective features for the input sentences, including separate models (Y. Chen et al., 2015) and joint inference models with event argument prediction (T. M. Nguyen & Nguyen, 2019). Among those deep neural networks, graph convolutional neural networks (GCN) (Kipf & Welling, 2017) have achieved state-of-the-art performance due to the ability to exploit the syntactic dependency graph to learn effective representations for the words (X. Liu et al., 2018; T. H. Nguyen & Grishman, 2018; Yan et al., 2019). However, two critical issues should be addressed to further improve the performance of such models.

First, given a sentence and a trigger candidate word, the hidden vectors induced by the current GCN models are not yet customized for the trigger candidate. As such, the trigger-agnostic representations in the GCN models might retain redundant/noisy information that is not relevant to the trigger candidate. As the trigger candidate is the focused word in the sentence, that noisy information might impair the performance of the ED models. To this end, we propose to filter the noisy information from the hidden vectors of GCNs so that only the relevant information for the trigger candidate is preserved. In particular, for each GCN layer, we introduce a gate, computed from the hidden vector of the trigger candidate, serving as the irrelevant information filter for the hidden vectors. Besides, as the hidden vectors in different layers of GCNs tend to capture the contextual information at different abstract levels, we argue that the gates for the different layers should also be regulated to exhibit such abstract representation distinction. Hence, we additionally introduce a novel regularization term for the overall loss function to achieve these distinctions for the gates.

Second, the current GCN models fail to consider the overall contextual importance scores of every word in the sentence. In previous GCN models, to produce the vector representation for the trigger candidate word, the GCN models mostly focus on the closest neighbors in the dependency graphs (X. Liu et al., 2018; T. H. Nguyen & Grishman, 2018). However, although the non-neighboring words might not directly carry useful context information for the trigger candidate word, we argue that their overall importance scores/rankings in the sentence for event prediction can still be exploited to provide useful training signals for the hidden vectors in ED. In particular, we propose to leverage the dependency tree to induce a **graph-based** importance score for every word based on its distance

to the trigger candidate. Afterward, we propose to incorporate such importance scores into the ED models by encouraging them to be consistent with another set of **model-based** importance scores that are computed from the hidden vectors of the models. Based on this consistency, we expect that graph-based scores can enhance the representation learning for ED. In our experiments, we show that our method outperforms the state-of-the-art models on the benchmark datasets for ED.

2.2 Model

2.2.1 Task Formulation.

The goal of ED consists of identifying trigger words (**trigger identification**) and classifying them for the event types of interest (**event classification**). Following the previous studies (T. H. Nguyen & Grishman, 2015), we combine these two tasks as a single multi-way classification task by introducing a *None* class, indicating non-event. Formally, given a sentence $X = [x_1, x_2, \ldots, x_n]$ of n words, and an index t ($1 \leq t \leq n$) of the trigger candidate x_t, the goal is to predict the event type y^* for the candidate x_t.

Our ED model consists of three modules: (1) Sentence Encoder, (2) GCN and Gate Diversity, and (3) Graph and Model Consistency.

2.2.2 Sentence Encoder.

We employ the pre-trained BERT (Devlin et al., 2019) to encode the given sentence X.

In particular, we create an input sequence of $[[CLS], x_1, \cdots, x_n, [SEP], x_t, [SEP]]$ where $[CLS]$ and $[SEP]$ are the two special tokens in BERT. The word pieces, which are tokenized from the sentence's words, are fed to BERT to obtain the hidden vectors of all layers. We concatenate the vectors of the top M layers to obtain the corresponding hidden vectors for each word piece, where M is a hyper-

parameter. Then, we obtain the representation of the sentence $E = \{e_1, \cdots, e_n\}$ in which the vectors e_i of x_i is the average of layer-concatenated vectors of its word pieces. Finally, we feed the embedding vectors in E to a bidirectional LSTM, resulting in a sequence of hidden vectors $h^0 = \{h_1^0, \cdots, h_n^0\}$.

2.2.3 GCN and Gate Diversity.

To apply the GCN model, we first build the sentence graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ for X based on its dependency tree, where $\mathcal{V}, \mathcal{E}$ are the sets of nodes and edges, respectively. $\mathcal{V}$ has n nodes, corresponding to the n words X. Each edge (x_i, x_j) in $\mathcal{E}$ amounts to a directed edge from the head x_i to the dependent x_j in the dependency tree. Following (Marcheggiani & Titov, 2017), we also include the opposite edges of the dependency edges and the self-loops in $\mathcal{E}$ to improve the information flow in the graph.

Our GCN module contains L stacked GCN layers (Kipf & Welling, 2017), operating over the sequence of hidden vectors h^0. The hidden vector h_i^l ($1 \leq i \leq n, 1 \leq l \leq L$) of the word x_i at the l-th layer is computed by averaging the hidden vectors of neighboring nodes of x_i at the $(l-1)$-th layer. Formally, h_i^l is computed as follow:

$$h_i^l = \text{ReLU}\left(W^l \sum_{(x_i, x_j) \in \mathcal{E}} \frac{h_j^{l-1}}{|\{x_j\}|}\right) \tag{2.1}$$

where W^l is a learnable weight of the GCN layer.

The major issue of the current GCN for ED is that its hidden vectors h_i^l are induced without special awareness of the trigger candidate x_t. This might result in irrelevant information (for the trigger word candidate) in the hidden vectors of GCNs for ED, thus hindering further performance improvement. To address this problem, we propose to filter that unrelated information by introducing a gate for each GCN layer. The vector g^l for the gate at the l-th layer is computed from the

embedding vector e_t of the trigger candidate:

$$g^l = \sigma(W_g^l e_t) \tag{2.2}$$

where W_g^l are learnable parameters for the l-th layer. Then, we apply these gates over the hidden vectors of the corresponding layer via the element-wise product, resulting in the filtered vectors:

$$m_i^l = g^l \circ h_i^l \tag{2.3}$$

As each layer in the GCN module has access to a particular degree of neighbors, the contextual information captured in these layers is expectedly distinctive. Besides, the gates for these layers control which information is passed through, therefore, they should also demonstrate a certain degree of contextual diversity. To this end, we propose to encourage the distinction among the outcomes of these gates once they are applied to the hidden vectors in the same layers. Particularly, starting with the hidden vectors h^l of of the l-layer, we apply the gates g^k (for all $(1 \leq k \leq L)$) to the vectors in h^l, which results in a sequence of filtered vectors:

$$\bar{m}_i^{k,l} = g^k \circ h_i^l \tag{2.4}$$

Afterward, we aggregate the filtered vectors obtained by the same gates using max-pooling:

$$\bar{m}^{k,l} = \text{max\_pool}(\bar{m}_1^{k,l}, \cdots, \bar{m}_n^{k,l}) \tag{2.5}$$

To encourage the gate diversity, we enforce vector separation between $\bar{m}^{l,l}$ with all the other aggregated vectors from the same layer l (i.e., $\bar{m}^{k,l}$ for $k \neq l$). As such, we introduce the following cosine-based regularization term $\mathcal{L}_{GD}$ (for Gate Diversity)

into the overall loss function:

$$\mathcal{L}_{GD} = \frac{1}{L(L-1)} \sum_{l=1}^{L} \sum_{k=l+1}^{L} \text{cosine}(\bar{m}^{l,l}, \bar{m}^{l,k}) \tag{2.6}$$

Note that the rationale for applying the gates g^k to the hidden vectors h^l for the gate diversity is to ground the control information in the gates to the contextual information of the sentence in the hidden vectors to facilitate meaningful context-based comparison for representation learning in ED.

2.2.4 Graph and Model Consistency.

As stated above, we seek to supervise the model using the knowledge from the dependency graph. Inspired by the contextual importance of the neighboring words for the event prediction of the trigger candidate x_t, we compute the **graph-based importance scores** $P = p_1, \cdots, p_n$ in which p_i is the negative distance from the word x_i to the trigger candidate.

In contrast, the model-based importance scores for each word x_i is computed based on the hidden vectors of the models. In particular, we first form an overall feature vector V_t that is used to predict the event type for x_t via:

$$V_t = [e_t, m_t^L, \text{max\_pool}(m_1^L, \cdots, m_n^L)] \tag{2.7}$$

In this work, we argue that the hidden vector of an important word in the sentence for ED should carry more useful information to predict the event type for x_t. Therefore, we consider a word x_i as more important for the prediction of the trigger candidate x_t if its representation m_i^L is more similar to the vector V_t. We estimate the **model-based important scores** for every word x_i with respect to the candidate x_t as follow:

$$q_i = \sigma(W^v V_t) \cdot \sigma(W^m m_i^L) \tag{2.8}$$

where W^v and W^m are trainable parameters.

Afterward, we normalize the scores P and $Q = \{q_1, \ldots, q_n\}$ using the softmax function. Finally, we minimize the KL divergence between the graph-based important scores P and the model-based importance scores Q by injecting a regularization term $\mathcal{L}_{ISC}$ (for the graph-model Importance Score Consistency) into the overall loss function:

$$\mathcal{L}_{ISC}(P, Q) = -\sum_{i=1}^{n} p_i \frac{p_i}{q_i} \tag{2.9}$$

To predict the event type, we feed V_t into a fully connected network with softmax function in the end to estimate the probability distribution $P(\hat{y}|X, t)$. To train the model, we use the negative log-likelihood as the classification loss

$$\mathcal{L}_{CE} = -\log P(y^*|X, t) \tag{2.10}$$

Finally, we minimize the following combined loss function to train the proposed model:

$$\mathcal{L} = \mathcal{L}_{CE} + \alpha\mathcal{L}_{GD} + \beta\mathcal{L}_{ISC} \tag{2.11}$$

where α and β are trade-off coefficients.

2.3 Experiments

Datasets: We evaluate our proposed model (called GatedGCN) on two ED datasets, i.e., ACE-2005 and Litbank. **ACE-2005** is a widely used benchmark dataset for ED, which consists of 33 event types. In contrast, **Litbank** is a newly published dataset in the literature domain, annotating words with two labels *event* and *none-event* (Sims et al., 2019). Hence, on Litbank, we essentially solve trigger identification with a binary classification problem for the words.

As the sizes of the ED dataset are generally small, the pre-processing procedures (e.g., tokenization, sentence splitting, dependency parsing, and selection of negative examples) might have a significant effect on the models' performance.

For instance, the current best performance for ED on ACE-2005 is reported by (S. Yang et al., 2019) (i.e., 80.7% F1 score on the test set). However, once we re-implement this model and apply it to the data version pre-processed and provided by the prior work (T. H. Nguyen & Grishman, 2015, 2018), we are only able to achieve an F1 score of 76.2% on the test set. As the models share the way to split the data, we attribute such a huge performance gap to the difference in data pre-processing that highlights the need to use the same pre-processed data to measure the performance of the ED models. Consequently, in this work, we employ the exact data version that has been pre-processed and released by the early work on ED for ACE-2005 in (T. H. Nguyen & Grishman, 2015, 2018) and for Litbank in (Sims et al., 2019).

The hyper-parameters for the models in this work are tuned on the development datasets, leading to the following selected values: one layer for the BiLSTM model with 128 hidden units in the layers, $L = 2$ for the number of the GCN layers with 128 dimensions for the hidden vectors, 128 hidden units for the layers of all the feed-forward networks in this work, and $5e$-5 for the learning rate of the Adam optimizer. These values apply for both the ACE-2005 and Litbank datasets. For the trade-off coefficients α and β in the overall loss function, we use $\alpha = 0.1$ and $\beta = 0.2$ for the ACE dataset while $\alpha = 0.3$ and $\beta = 0.2$ are employed for Litbank. Finally, we use the case $\text{BERT}_{\text{base}}$ version of BERT and freeze its parameters during training in this work. To obtain the BERT representations of the word pieces, we use $M = 12$ for ACE-2005 and $M = 4$ for Litbank (Sims et al., 2019).

Results: We compare our model with two classes of baselines on ACE-2005. Note that these baselines use the same pre-processed data as ours.

The first class includes the models with non-contextualized embedding:

– **CNN**: a CNN model (T. H. Nguyen & Grishman, 2015)

– **NCNN**: non-consecutive CNN model: (T. H. Nguyen & Grishman, 2016)

– **GCN-ED**: a GCN model (T. H. Nguyen & Grishman, 2018)

The second class of baselines concerns the models with the contextualized embeddings. These models currently have the best-reported performance for ED on ACE-2005. Note that as these works employ different pre-processed versions of ACE-2005, we re-implement the models and tune them on our dataset version for a fair comparison.

– **DMBERT**: a model with dynamic pooling (H. Wang et al., 2019)

– **BERT+MLP**: a MLP model with BERT (S. Yang et al., 2019).

For Litbank corpus, we use the following baselines reported in the original paper (Sims et al., 2019):

– **BiLSTM**: a BiLSTM model with Word2Vec.

– **BERT+BiLSTM**: a BiLSTM model with BERT.

– **DMBERT** a model with dynamic pooling (H. Wang et al., 2019).

Table 7 presents the performance of the models on the ACE-2005 test set. This table shows that GatedGCN outperforms all the baselines with a significant improvement of 1.4% F1-score over the second-best model BERT+MLP. In addition, Table 8 shows the performance of the models on the Litbank test set. As can be seen, the proposed model is better than all the baseline models with

Model	Precision	Recall	Fscore
CNN	71.8	66.4	69.0
NCNN	-	-	71.3
GCN-ED	77.9	68.8	73.1
DMBERT	79.1	71.3	74.9
BERT+MLP	77.8	74.6	76.2
BERT+GCN	80.3	73.0	76.5
GatedGCN	78.8	76.3	**77.6**

Table 7. Performance on the ACE-2005 test set.

0.6% F1-score improvement over the state-of-the-art model BERT+BiLSTM. These improvements are significant on both datasets ($p < 0.05$), demonstrating the effectiveness of GatedGCN for ED.

Model	Precision	Recall	Fscore
BiLSTM	70.4	60.7	65.2
+ document context	74.2	58.8	65.6
+ sentence CNN	71.6	56.4	63.1
+ subword CNN	69.2	64.8	66.9
DMBERT	65.0	76.7	70.4
BERT+BiLSTM	75.5	72.3	73.9
BERT+GCN	71.0	76.3	73.6
GatedGCN	69.9	79.8	**74.5**

Table 8. Performance on the Litbank test set.

Ablation Study: The proposed model involves three major components: (1) the **Gates** to filter irrelevant information, (2) the Gate **Diversity** to encourage contextual distinction for the gates, and (3) the **Consistency** between graph and model-based importance scores. Table 9 reports the ablation study on the ACE-2005 development set when the components are incrementally removed from the full model (note that eliminating **Gate** also removes **Diversity** at the same time). As can be seen, excluding any component results in significant performance reduction,

Model	Precision	Recall	Fscore
GatedGCN (full)	76.7	70.5	**73.4**
-Diversity	78.5	67.0	72.3
-Consistency	80.5	64.7	71.7
-Diversity -Consistency	79.0	63.0	70.1
-Gates	77.8	65.3	71.3
-Gates -Consistency	83.0	62.5	71.0

Table 9. Ablation study on the ACE-2005 dev set.

clearly testifying to the benefits of the three components in the proposed model for ED.

Importance Score Visualization: In order to further demonstrate the operation of the proposed model GatedGCN for ED, we analyze the model-based importance scores for the words in test set sentences of ACE-2005 that can be correctly predicted by GatedGCN, but leads to incorrect predictions for the ablated model "-Gate-Consistency" in Table 9 (called the GatedGCN-successful examples). In particular, Figure 2 illustrates the model-based importance scores for the words in the sentences of several GatedGCN-successful examples. Among others, we find that although the trigger words are directly connected to several words (including the irrelevant ones) in these sentences, the **Gates**, **Diversity**, and **Consistency** components in GatedGCN help to better highlight the most informative words among those neighboring words by assigning them larger importance scores. This enables the representation aggregation mechanism in GCN to learn better hidden vectors, leading to improved performance for ED in this case.

2.4 Related Work

Prior studies on ED involve handcrafted feature engineering for statistical models (Ahn, 2006; Hong et al., 2011; Ji & Grishman, 2008; Mitamura et al., 2015) and deep neural networks, e.g., CNN (Y. Chen et al., 2015, 2018; T. H. Nguyen &

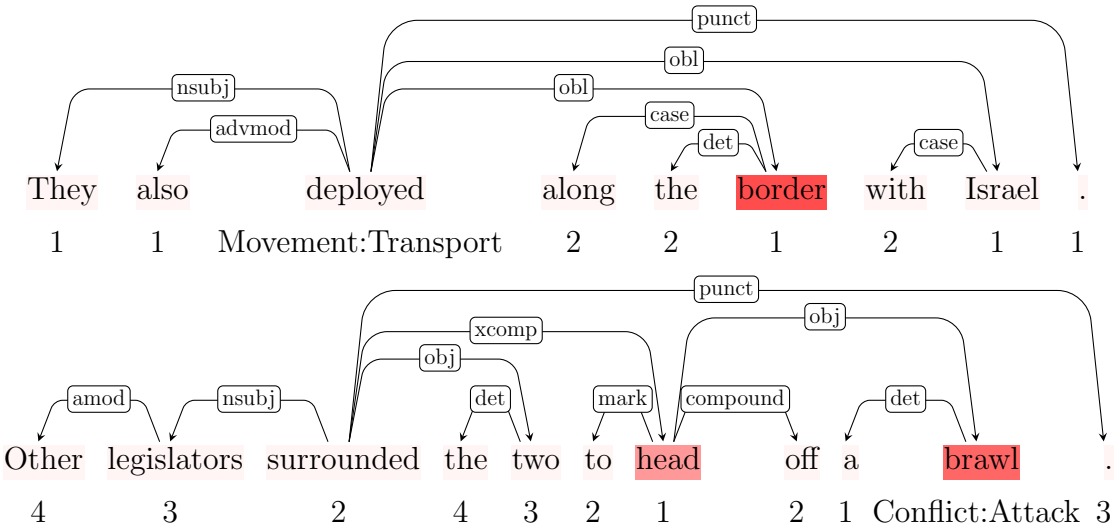

Figure 2. Visualization of the model-based importance scores computed by the proposed model for several GatedGCN-successful examples. The words with bolder colors have larger importance scores in this case. Note that the golden event types "*Movement:Transport*" and "*Conflict:Attack*" are written under the trigger words in the sentences. Also, below each word in the sentences, we indicate the number of words along the path from that word to the trigger word (i.e., the distances used in the graph-based importance scores).

Grishman, 2015; T. H. Nguyen, Meyers, & Grishman, 2016g), RNN (Feng et al., 2016; Jagannatha & Yu, 2016; T. H. Nguyen, Cho, & Grishman, 2016), attention mechanism (Y. Chen et al., 2018; S. Liu, Chen, Liu, & Zhao, 2017), contextualized embeddings (S. Yang et al., 2019), and adversarial training (H. Wang et al., 2019). The last few years witness the success of graph convolutional neural networks for ED (X. Liu et al., 2018; T. H. Nguyen & Grishman, 2018; Pouran Ben Veyseh, Nguyen, & Dou, 2019; Yan et al., 2019) where the dependency trees are employed to boost the performance. However, these graph-based models have not considered representation regulation for GCNs and exploiting graph-based distances as we do in this work.

2.5 Summary

In summary, the main contribution of this chapter includes:

- We addressed the noisy information from the hidden vectors of the graph convolutional neural network for ED by filtering out irrelevant information for the candidate event trigger. In particular, we introduce a gate for each layer of the graph convolutional neural network. The gate kernel is computed from the event trigger candidate to customize the filter for each event trigger.

- We also proposed a novel regularization term to facilitate gate diversity between gates of different layers.

- We proposed a method to incorporate the syntactic importance score based on the distances on the dependency graph to enrich the representation learning of the model. To do that, we enforce the importance score distribution similarities between the graph-based importance score and model-generated importance score.

- Our extensive experiments on two benchmark datasets (ACE-05 and Litbank) show that our methods improve the performance of the GCN-based model.

While the proposed method is effective in enriching the representation in graph convolutional neural networks, these models under supervised learning can not work with new event types. In the next chapter, we present our attempt to extend event extraction into new event types under the few-shot learning scheme. The few-shot learning model has to generalize for any new event types that using training signal from the training data is not sufficient. Hence, we introduce a transfer learning method to improve the model not only few-shot learning but also supervised learning.

CHAPTER III

GRAPH LEARNING REGULARIZATION AND TRANSFER LEARNING FOR FEW-SHOT EVENT DETECTION

This chapter includes the materials from a published paper "*Viet Dac Lai, Minh Van Nguyen, Thien Huu Nguyen, and Franck Dernoncourt.* **Graph learning regularization and transfer learning for few-shot event detection.** *In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 2172-2176. 2021.*"

As the first author of this paper, Viet was responsible for the development, evaluation, and writing. Minh, Franck, and Thien provided meaningful discussion and analysis. Franck and Thien have put on editorial writing for the paper submission. The paper was revised to comply with the dissertation format and purposes.

This chapter addresses the poor generalization of few-shot learning models for event detection (ED) using transfer learning and representation regularization. In particular, we propose to transfer knowledge from open-domain word sense disambiguation into few-shot learning models for ED to improve their generalization to new event types. We also propose a novel training signal derived from dependency graphs to regularize the representation learning for ED. Moreover, we evaluate few-shot learning models for ED with a large-scale human-annotated ED dataset to obtain more reliable insights into this problem. Our comprehensive experiments demonstrate that the proposed model outperforms state-of-the-art baseline models in the few-shot learning and supervised learning settings for ED.

3.1 Introduction

Event Detection (ED) is a natural language processing (NLP) task that detects event triggers/mentions (i.e., the most important words to clearly express an event) and categorizes them into a set of predefined event types. For instance, given the following sentence, an ED model should detect the word *skirmish* as an event trigger and classify it as *CONFLICT-ATTACK*:

*"Fans **skirmish** ahead of the match in Marseille on Saturday."*

Existing works have mostly solved ED in the supervised learning setting (Y. Chen et al., 2015; Feng et al., 2016; T. H. Nguyen & Grishman, 2018; S. Yang et al., 2019). In real-world applications, a major problem of these supervised ED models is the poor transferability to new event types (L. Huang et al., 2018). As such, the predictions of trained models are limited to predefined event types, thereby failing to extract event triggers of new types. Recent studies address this issue by formulating ED as a low-shot learning problem in low-resource conditions, including zero-shot learning (L. Huang et al., 2018) and few-shot learning (FSL) (V. D. Lai, Nguyen, & Dernoncourt, 2020). These methods enable models to effectively extend the operation to new event types, for which no or a few training samples are annotated. In this work, we focus on the few-shot learning setting, aiming to address three issues in the existing FSL methods for ED.

First, current models in few-shot learning for ED are only evaluated on datasets with small numbers of event types. For instance, recent few-shot learning studies (V. D. Lai, Nguyen, & Dernoncourt, 2020) mainly use the popular ACE 2005 dataset that only contains 33 event types (Grishman, Westbrook, & Meyers, 2005). This makes the reported performance in those prior work less reliable as the utilized datasets cannot cover a wide range of possible event types to better

estimate the generalization. Besides, due to the small number of event types, prior FSL work for ED has to use the same event types for the development and test datasets (V. D. Lai, Nguyen, & Dernoncourt, 2020), thereby violating the requirement of disjoint event types for the training, testing, and development data in FSL and leading to an unrealistic setting for this problem. To address this issue, this work conducts the first FSL research for ED where the evaluation is performed on a human-annotated ED dataset with a large number of event types to enable more realistic and reliable performance. In particular, we employ a recently released event extraction dataset RAMS, *Roles Across Multiple Sentences* (Ebner et al., 2020) (with 139 event types), to extensively evaluate various FSL models for ED in this work.

The second issue involves the failure to exploit knowledge from ED-related datasets/tasks to advance the generalization for the models (V. D. Lai, Nguyen, & Dernoncourt, 2020). As such, our intuition is that FSL models can generalize better to new event types if they are augmented with knowledge (knowledge transferring) from datasets with a large number of event types (ideally all the possible event types).

Motivated by the prior work on supervised ED (W. Lu & Nguyen, 2018), we resort to Semcor, a human-annotated dataset for word sense disambiguation (WSD), to obtain the knowledge about open-domain event types and transfer it to FSL models for ED. Besides the high quality of the data (due to the human annotation), Semcor provides the annotations for a large number of word senses in WordNet that can cover a variety of event types and potentially improve the type generalization of the augmented FSL models (W. Lu & Nguyen, 2018). To our knowledge, this is the first work to explore transfer learning for FSL in ED.

Finally, to further improve the performance of FSL models for ED, we propose a novel regularization mechanism to produce better representation vectors. Our mechanism differentiates two types of words in a sentence for an event trigger, i.e., relevant words and irrelevant words. On the one hand, we argue that the representation vector for the event trigger should be computed mainly based on the relevant words. On the other hand, we expect that the irrelevant words can also provide useful training signals for ED models by introducing constraints to force these words to not contribute significantly to the learned hidden vectors. As such, in addition to inducing hidden vectors based on the relevant words, we propose to obtain representation vectors from every word in the sentence (i.e., including both relevant and irrelevant words). To minimize the contribution of the irrelevant words, we then introduce a regularization term to enforce the similarity between the hidden vectors from the relevant words and the whole sentence. Our extensive experiments demonstrate the effectiveness of the proposed techniques for ED, leading to state-of-the-art performance in both FSL and supervised learning settings.

3.2 Background

In few-shot learning, we are given a set of labeled data $\mathcal{D}^{train}$ corresponding to a set of classes $\mathcal{Y}^{train}$. A learning model has to exploit knowledge from this data so later it can predict on a completely new set of classes $\mathcal{Y}^{test}$ (with the labeled data set $\mathcal{D}^{test}$), in which only a few annotated samples (e.g., 5 or 10) is provided for each new class. As such, the model is trained over a set of classes $\mathcal{Y}^{train}$, then it is tested on $\mathcal{Y}^{test}$ which is disjoint from $\mathcal{Y}^{train}$.

Few-Shot Learning To emulate the above setting, we follow the conventional *episodic training* (Vinyals, Blundell, Lillicrap, & Wierstra, 2016) to

sample training tasks. In each training episode (i.e., training iteration), we sample a subset of N classes $\mathcal{Y}$ from $\mathcal{Y}^{train}$. For each class $t_i \in \mathcal{Y}$, we sample $K + Q$ examples of which K examples serve as training data, and Q examples are used for testing data. Gathering training data and testing data for all classes, we have a *meta-training set* and a *meta-testing set*. In the literature, they are also called *support set* and *query set* respectively. In each training episode, the parameters of a learner are updated based on the loss over the query set.

Once we have a meta-trained model, the same episodic sampling process is employed multiple times over the $\mathcal{D}^{test}$ to evaluate how quickly the model adapts to a brand-new set of classes. In particular, we first sample N classes from $\mathcal{Y}^{test}$, then, we sample K examples per class as the support set and Q examples per class as the query set. To clarify, the *N-way K-shot* few-shot learning setting refers to the task of making prediction over the query set, given a support set of $N \times K$ examples **during meta-testing**.

Framework Following prior works in ED (T. H. Nguyen & Grishman, 2015), we add an additional *NULL* class in every task to indicate a *not-an-event* class. Thus, the FSL ED problem can be formulated as *N+1-way K-shot* few-shot classification problem. We employ the following general metric-based framework for FSL with two following components:

Instance Encoder: Given a sentence of N words $s = \{w_1, .., w_N\}$ and the position a of the trigger word $w_a \in s$ for some example/instance. We employ a deep neural network, denoted by a function f, to encode the instance into a fixed-dimension representation vector $f(s, a) \in R^d$.

Few-shot Classifier: A prototype is a representative vector c for each class appearing in the support set (called the prototype vector for the class). It can

be an average (Snell et al., 2017) or a weighted sum with query-based attention weights (T. Gao, Han, Liu, & Sun, 2019) of vectors from the support set. Then, by computing the distance between the representation vector of a query instance $q = (s^q, a^q, t^q)$ and the prototype vectors, we can obtain a distance-based distribution over the possible classes in the current episode for q:

$$P(y = t^j | q, \mathcal{S}) = \frac{e^{-D(f(s^q, a^q), c^j)}}{\sum_{k=1}^{N+1} e^{-D(f(s^q, a^q), c^k)}} \tag{3.1}$$

where D is a distance function (e.g. Euclidean distance (Snell et al., 2017), cosine similarity (Vinyals et al., 2016)), c^k is the prototype vector for the k-th class (Snell et al., 2017). Given this distribution, the loss function L_{FSL} to train the FSL models is the negative log-likelihood computed for each query instance q:

$$L_{FSL} = -\log P(y = t^q | q, S) \tag{3.2}$$

3.3 Proposed Model

Instance Encoder To differentiate between relevant words and irrelevant words, the instance encoder component in our model first focuses on relevant words in sentences to achieve this goal. As such, to identify the relevant words for an event trigger candidate in a sentence, we rely on the structure of the arguments of the trigger candidate where arguments have been shown to provide useful information to identify the event trigger (S. Liu et al., 2017). In particular, we use the dependency parsing tree and their argument-related dependency paths to compute the representation vector for the trigger candidate. Given the sentence $s = w_1, w_2, \ldots, w_N$ and the trigger position a, we first embed s using the BERT model (Devlin et al., 2019) to produce a representation vector h_i^0 for each word $w_i \in s$. Next, to induce hidden representation using the relevant words for the trigger, we build a pruned dependency graph following two steps:

Given a sentence, we first obtain its dependency tree. Then we convert it into an undirected graph by eliminating all directions and inserting self-loops. This process results in a full dependency graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.

Having a list of all entity mentions in the sentence, we find all the paths from the trigger candidate to the entity mention words. Then we eliminate all the edges of $\mathcal{G}$ that do not belong to any of the above paths, leading to a pruned dependency graph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$. Note that $\mathcal{G}$ and $\mathcal{G}'$ involve the same set of nodes for the words in the input sentence. For convenience, let A and A' be the adjacent matrices of the graphs $\mathcal{G}$ and $\mathcal{G}'$, respectively. In the next step, given the graphs $\mathcal{G}$ and $\mathcal{G}'$, we seek to induce abstract representation vectors for the nodes using GCNs (Kipf & Welling, 2017). As such, the GCN model in our work involves several hidden layers in which the representation vector of the i-th node/word at the l-th layer is computed as follows:

$$h_i^l(\mathcal{G}^{(\cdot)}) = \text{ReLU}(d_i^{-1} \sum_{j=1}^{N} A_{ij}^{(\cdot)} W^l h_j^{l-1} + b^l) \tag{3.3}$$

where $(\cdot)$ indicate which graph (i.e., $\mathcal{G}$ or $\mathcal{G}'$) to be used, $d_i = \sum_{j=1}^{N} A_{ij}^{(\cdot)}$ is the degree of the node w_i, W^l, b^l are learnable parameters (Kipf & Welling, 2017), and $ReLU$ is the Rectified Linear Unit.

Finally, to embed the trigger candidate w_a into a representation vector, we concatenate the hidden vectors of the trigger candidate from BERT h_a^0 and all GCN layers $h_a^k(\mathcal{G}')(k > 0)$ (based on $\mathcal{G}'$), then feed it to a one-layer feed-forward neural network:

$$f(s, a) = v(\mathcal{G}') = W \tanh([h_a^0, h_a^1(\mathcal{G}'), \cdots, h_a^L(\mathcal{G}')]) + b \tag{3.4}$$

where W, b are trainable parameters; L is the number of GCN layers. For convenience, the encoder with BERT and GCN as in Equation 3.4 is called the

BERTGCN model to contrast with the **BERTMLP** model where $f(s, a)$ is only set to $Wh_a^0 + b$ (i.e., not using GCN model). Note that **BERTMLP** is also one of the current state-of-the-art models for ED (V. D. Lai, Nguyen, & Nguyen, 2020b).

Graph-based Regularization Our target is to regulate the representation learning based on dependency graphs, aiming to eliminate the contribution of irrelevant words. By introducing the pruned graph, we have partially achieved this goal. However, irrelevant words might still contribute to the representation vectors in the model due to the BERT encoder that is run over the entire input sentence. To further constrain the contribution of irrelevant words for representation learning, we seek to impose a similarity requirement over the representation vectors obtained via the pruned tree $\mathcal{G}'$ and the full tree $\mathcal{G}$. In other words, we ensure that adding irrelevant words in the pruned tree does not change representation vectors significantly.

To implement this idea, given the full dependency graph $\mathcal{G}$ and the pruned graph $\mathcal{G}'$, we first obtain two representation vectors V and V' for the input sentence s based on $\mathcal{G}$ and $\mathcal{G}'$ respectively via:

$$\begin{aligned} m^l(\mathcal{G}^{(\cdot)}) &= \max_i(h_1^l(\mathcal{G}^{(\cdot)}), \cdots, h_N^l(\mathcal{G}^{(\cdot)})) \\ V^{(\cdot)} &= \text{concat}(m^1(\mathcal{G}^{(\cdot)}), \cdots, m^L(\mathcal{G}^{(\cdot)})) \end{aligned} \tag{3.5}$$

In the next step, to limit the contribution of irrelevant words, we enforce the similarity between V and V' by adding the KL divergence, i.e., $L_{GRAPH} = KL(\sigma(V), \sigma(V'))$, between them into the overall loss function for minimization (σ is the softmax function to obtain distributions for the KL divergence).

Transfer Learning Our goal is to improve the generalization of the FSL ED model by transferring open-domain knowledge from WSD into the FSL ED model. Prior work on transfer learning for ED employs a matching

method (W. Lu & Nguyen, 2018) which presents two separate neural networks with identical architecture and different parameters for ED and WSD. In each training iteration, a task is sampled and the model for that task is trained (W. Lu & Nguyen, 2018) using the cross-entropy loss (called **ALTERNATE** training). In addition, transfer learning is achieved by introducing an auxiliary loss to enforce the similarity between hidden vectors generated by the two models on the same sentences. However, directly applying this method for FSL might result in a drastic reduction of performance. First, the vectors generated by the two models might be mismatched due to the semantic difference of the tasks. Second, a significant difference between the learning speed of the two models requires manual calibration of learning rates during the training, leading to suboptimal solutions (Guo, Che, Wang, Liu, & Xu, 2016; W. Lu & Nguyen, 2018). This learning speed gap might be even more pronounced in FSL as FSL tends to converge faster than supervised learning. Finally, sharing an identical architecture might limit the robustness of WSD and ED models because the best model for a particular task cannot be employed. Therefore, we propose to separately pre-train the WSD model from the ED model that allows the WSD model to inherit the best WSD architecture to produce effective representations for sentences upfront. The ED model is trained afterward, acquiring the transferred knowledge from the WSD model. In this way, the learning rate gap issue is also automatically avoided to enhance the ED performance.

Formally, we employ two separate deep neural networks whose encoders are denoted as f_{ed} and f_{wsd} for ED and WSD, respectively. We have two datasets D_{ed} and D_{wsd}:

$$D_{ed} = \{(s_i^{ed}, a_i^{ed}, t_i^{ed})\}$$

$$D_{wsd} = \{(s_j^{wsd}, a_j^{wsd}, t_j^{wsd})\}$$

where the notation of (s, a, t) are similar for two tasks (W. Lu & Nguyen, 2018). They stand for a sentence s, the position a of a candidate anchor word in s, and the golden label t (i.e., an event type in ED and a word sense in WSD).

First, we train a WSD model using WSD data. The parameters of the trained WSD model will be fixed and its knowledge will be later transferred to the ED model:

$$f_{wsd}^* \leftarrow \operatorname*{argmin}_{f_{wsd}} \sum_{(s,a,t) \in D_{wsd}} L(f_{wsd}(s, a), t) \tag{3.6}$$

Second, we train the ED model. In each ED training iteration, we sample an instance (s, a, t) from either D_{ed} or D_{wsd}, then feed it to the two model encoders to get two corresponding representations v^{ed} and v^{wsd} (using Equation 3.4). Finally, transfer learning regularization from WSD to ED is performed by minimizing the KL divergence between v^{ed} and v^{wsd} (i.e., to promote the representation similarity over the same example (s, a)):

$$L_{WSD} = KL(\sigma(f_{ed}(s, a)), \sigma(f_{wsd}^*(s, a))) \tag{3.7}$$

Finally, to train the proposed model, we minimize the combination of the proposed losses with α, β as two trade-off coefficients:

$$L = L_{FSL} + \alpha L_{WSD} + \beta L_{GRAPH} \tag{3.8}$$

3.4 Evaluation

Datasets: We evaluate our methods on two ED datasets. First, as presented in the introduction, to enable a more realistic evaluation for FSL ED models, we employ the RAMS dataset (recently released by (Ebner et al., 2020)) that provides human annotation for a large number of event types, involving 9124 examples/triggers for 139 event types. As RAMS is originally divided (for

train/dev/test data portions) for traditional supervised learning, we first combine the data portions and re-split RAMS based on event types to facilitate FSL evaluation.

Second, to further evaluate the ED models in the traditional supervised learning setting, we utilize the widely used ACE-2005 dataset (Walker et al., 2006) that annotates 33 event subtypes. As discussed in (V. D. Lai, Nguyen, & Nguyen, 2020b), using the same data preprocessing is crucial for a fair comparison between methods on ACE-2005. To this end, we use the exact data split (i.e., train/dev/test) and data preprocessing provided by (V. D. Lai, Nguyen, & Nguyen, 2020b), the current state-of-the-art ED model for model evaluation on ACE-2005 in this work. Finally, we employ the **Semcor** dataset for WSD (Miller, Chodorow, Landes, Leacock, & Thomas, 1994) (annotated with word senses in WordNet 3.0 (Miller, 1995)) to pre-train the WSD model for our transfer learning component.

Hyperparameters: We select the hyper-parameters for the proposed model based on the performance on the development set of RAMS. We employ the BERT-base-cased version of BERT and use the hidden vectors of the top $M = 4$ layers for the representation vectors h_i^0. For the GCN model, we stack $L = 2$ GCN layers; each has 512 hidden units. The dimensionality d of the representation vectors $f(s, a)$ for instances is set to 128. We use the state-of-the-art BERT-based WSD model in (Hadiwinoto, Ng, & Gan, 2019) to pre-train the WSD model for transfer learning in this work. Our FSL models are trained in 6000 episodes and tested with 500 episodes. The learning rate for FSL models is set to $2e10^{-4}$ with the Adam optimizer.

FSL setting: We evaluate all the models using the 5+1-way 5-shot FSL setting. As the previous study has observed that training FSL setting with a larger

Model	BERTMLP			BERTGCN		
	Precision	**Recall**	**Fscore**	**Precision**	**Recall**	**Fscore**
Prototypical	66.5	70.1	68.2	69.9	72.4	71.0
InterIntra	67.6	70.9	69.2	71.1	73.7	72.4
GraphTransfer	**68.9**	**70.6**	**69.7**	**71.9**	**74.7**	**73.2**

Table 10. Performance of FSL models with the 5+1-way 5-shot FSL on the RAMS test set.

N^{train} results in better performance during testing (Snell et al., 2017), we sample $N^{train} = 20$ event subsubtypes in each training batch while still keeping $N^{test} = 5$ during test time.

Baseline: We consider two classes of baseline methods for FSL ED. The first class involves FSL methods that have been designed for other NLP tasks, including matching networks (Vinyals et al., 2016), prototypical networks (Snell et al., 2017), hybrid-attention prototypical networks (T. Gao et al., 2019), and relation networks (Sung et al., 2018). Among these methods, the prototypical network (called **Prototypical**) produces the best performance in our experiments and we will use it to represent the first class of baselines in this work. Note that the selection of prototypical networks will also determine the distance function D in Equation 3.1. Second, we also utilize **InterIntra**, the current state-of-the-art technique for FSL ED in (V. D. Lai, Nguyen, & Dernoncourt, 2020) as the baseline. Finally, we examine both **BERTMLP** and **BERTGCN** as the instance encoders for FSL models in this work.

3.4.1 Few-Shot Learning Evaluation.

Table 10 compares the baseline FSL models without proposed method (called GraphTransfer) on the RAMS test set. The first observation is that the GCN-based encoder BERTGCN is significantly better than the non-graph encoder BERTMLP across different FSL methods, thus highlighting the benefits of GCN

for FSL ED. More importantly, the proposed model significantly outperforms all the baseline models with $p < 0.05$. The consistent improvement for both instance encoder architectures demonstrates the effectiveness of the proposed FSL models for ED in this work.

3.4.2 Ablation study.

Our proposed method GraphTransfer involves two main components: (i) transferring learned knowledge from pre-trained WSD task (**WSD**) and (ii) graph-based regularization (**GRAPH**). We also propose the fix training strategy, called **FIX**, to pre-train the WSD model for transfer learning (i.e., in contrast to the ALTERNATE method in (W. Lu & Nguyen, 2018)), and the use of relevant words derived from the pruned graph for prediction (**Prune**). To analyze the contribution of these components, we incrementally remove these components from the full model and reevaluate the remaining models. Note that by eliminating the **WSD** component, we also exclude the **FIX** strategy due to their dependency.

Model	**Precision**	**Recall**	**Fscore**
GraphTransfer (full)	**71.9**	**74.7**	**73.2**
-WSD	71.4	74.2	72.7
-GRAPH	70.8	73.5	72.1
-GRAPH-WSD	69.9	72.4	71.0
-GRAPH-WSD-Prune	69.1	72.6	70.7
-FIX (using ALTERNATE)	71.8	73.3	72.5

Table 11. Ablation study on RAMS dataset

Table 11 presents the performance of 5+1-way 5-shot few-shot learning on RAMS. As shown in the table, eliminating either **WSD** or **GRAPH** significantly hurts the performance of the model. In addition, the performance is further reduced when the full dependency graph is used to compute the instance representations (i.e., instead of using the pruned graph equation 1.1).

Finally, excluding the **FIX** training strategy in transfer learning (i.e., using ALTERNATE in (W. Lu & Nguyen, 2018) instead) also leads to significantly reduced performance.

3.4.3 Supervised Learning Evaluation.

We compare our proposed model against current state-of-the-art models for ED in the supervised learning setting on the ACE-2005 dataset, including **DMBERT** (H. Wang et al., 2019) (a BERT-based model with dynamic pooling), **BERTGCN** (as presented above), and **BERTMLP** and **Gated-GCN** (V. D. Lai, Nguyen, & Nguyen, 2020b). Note that **Gated-GCN** also uses BERT and it is the current state-of-the-art ED model for supervised learning with our dataset setting on ACE-2005. For completeness, we also provide Gate-GCN 's performance on RAMS in the supervised learning setting using its original data split.

Model	**RAMS**			**ACE-2005**		
	Precision	**Recall**	**Fscore**	**Precision**	**Recall**	**Fscore**
DMBERT	62.6	44.0	51.7	79.1	71.3	74.9
BERTMLP	62.4	49.3	55.0	77.8	74.6	76.2
BERTGCN	**66.5**	59.0	62.5	80.2	74.8	77.4
Gated-GCN	64.8	64.5	64.7	78.8	76.3	77.6
GraphTransfer	66.3	**65.8**	**66.1**	**80.3**	**78.0**	**79.1**

Table 12. Supervised learning performance.

Result: Table 12 reports the performance of the models. It is clear from the table that the proposed model significantly outperforms all baseline models with large margins over the current best model, i.e., 3.6% on RAMS, and 1.5% on ACE-2005, thereby further confirming the effectiveness of the proposed model for ED.

3.5 Related Work

Early studies have addressed ED via the supervised learning setting (Ahn, 2006; Y. Chen et al., 2015; Feng et al., 2016; Hong et al., 2011; Ji & Grishman, 2008; Liao & Grishman, 2010; M. V. Nguyen, Lai, & Nguyen, 2021; T. H. Nguyen, Cho, & Grishman, 2016; T. H. Nguyen, Fu, Cho, & Grishman, 2016; T. H. Nguyen & Grishman, 2015, 2018). Extending ED to unseen event types is an emerging direction for which several approaches have been proposed, including bootstrapping (R. Huang & Riloff, 2012), self-training (Liao & Grishman, 2011), zero-shot learning (L. Huang et al., 2018), distant supervision (Y. Chen et al., 2018; Tong et al., 2020), and FSL (V. D. Lai, Dernoncourt, & Nguyen, 2020; V. D. Lai, Nguyen, & Dernoncourt, 2020). FSL promotes effective learning from small numbers of examples for new types. The major approaches include metric learning (Deng et al., 2020; T. Gao et al., 2019; Snell et al., 2017; Sung et al., 2018; Vinyals et al., 2016) and meta-learning (Finn, Abbeel, & Levine, 2017; K. Lee, Maji, Ravichandran, & Soatto, 2019). Finally, several studies have employed transfer learning for few-shot learning (Bao, Wu, Chang, & Barzilay, 2020; Shalyminov, Lee, Eshghi, & Lemon, 2019); however, none of them has explored transfer learning for FSL ED as we do.

3.6 Summary

The contribution of this chapter includes:

- We present how transferring open-domain knowledge from word sense disambiguation and regulating representation based on pruned dependency graphs can improve few-shot learning for ED on large-scale datasets.

- Our proposed model achieves state-of-the-art performance on both few-shot learning and supervised learning on two ED datasets.

While the method in this chapter has improved the performance of the ED models, these models under the few-shot learning setting suffer from noisy sampling appearing in episodical training. In the next chapter, we address the poor sampling in episodical training, particularly for ED tasks. Then, we propose a method to help the model mitigate the issue, creating a more robust few-shot classifier.

CHAPTER IV

LEARNING PROTOTYPE REPRESENTATIONS ACROSS FEW-SHOT TASKS FOR EVENT DETECTION

This chapter contains materials from the published paper *Lai, Viet, Franck Dernoncourt, and Thien Huu Nguyen.* **Learning Prototype Representations Across Few-Shot Tasks for Event Detection.** *In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 5270-5277. 2021.*

As the first author of this paper, Viet was responsible for the development, evaluation, and writing. Franck and Thien provide meaningful discussion and editorial revision of the submitted paper. The paper was revised to comply with the format and the purposes of this dissertation.

In this chapter, we continue to address the issues of the few-shot learning models for the ED problem. In particular, we address the sampling bias and outlier issues in few-shot learning for event detection. To overcome it, we propose to model the relations between training tasks in episodic few-shot learning by introducing cross-task prototypes. We further propose to enforce prediction consistency among classifiers across tasks to make the model more robust to outliers. Our extensive experiment shows a consistent improvement on three few-shot learning datasets for ED. The findings suggest that our model is more robust when labeled data of novel event types is limited.

4.1 Introduction

In Information Extraction, Event Detection (ED) is an important task that aims to identify and classify event triggers of predefined event types in text (Walker et al., 2006). Event triggers are words/phrases that most clearly indicate

the occurrence of events. For example, an event detector should recognize the word *homicide* in the following sentence as a trigger word of event type *life.die.death-caused-by-violent-events*:

"*...the medical examiner believed the manner of death was an accident rather than a* **homicide**."

Typical ED systems follow a supervised learning scheme that requires a large amount of labeled data for each predefined event type (Y. Chen et al., 2015; Ji & Grishman, 2008; M. V. Nguyen, Lai, & Nguyen, 2021; T. H. Nguyen & Grishman, 2015). Unfortunately, this requirement is usually too costly to achieve in real applications where novel event types emerge and only a few examples are available (L. Huang et al., 2018). As such, an ED model should be prepared to extract triggers of novel event types (i.e., beyond those provided in the training data) for which only a few examples are provided. This learning schema is known as **Few-Shot Learning** (FSL) for ED.

To emulate the learning from a few examples in ED, N-way K-shot episodic training is often used to exploit existing datasets (Deng et al., 2020; V. D. Lai, Dernoncourt, & Nguyen, 2020; V. D. Lai, Nguyen, Nguyen, & Dernoncourt, 2021; V. D. Lai, Nguyen, & Dernoncourt, 2020). In each training iteration, a small subset (i.e. **support set**) of N event types with K examples per type is sampled from the training data. Unfortunately, the sample size is so small ($K \in [1, 10]$) that the FSL models might suffer from sample bias, thus hindering the generalization to novel event types.

The prototypical network is a popular metric-based few-shot learning model (Snell et al., 2017) that has been explored for FSL ED (Deng et al., 2020; V. D. Lai, Nguyen, & Dernoncourt, 2020). It introduces a prototype vector for

each event type by averaging the representations of the instances of that type. A non-parametric classifier then predicts the event type of a query instance based on its distances from the prototypes (Snell et al., 2017). Hence, an outlier in the support set might significantly change the prototypes and flip the label of the query instance. In addition, in ED, a NULL class is introduced to represent non-eventive mentions. This type covers every domain and every surface form except the relevant event types. Thus, this unbounded class might also present a great source of outliers for the support set.

In this work, we mitigate the effects of poor sampling and outliers by modeling cross-task relations. First, we propose to augment the support data of the current task with those from prior tasks which essentially helps increase the population of the current support set. Therefore, it can mitigate the sample bias in the support set. Second, the averaging in the prototypical network allows outliers to contribute equally to the prototype representation. We propose to use soft attention to select the most related data samples as well as reduce the contribution of the outliers to the prototype representation. Third, an FSL model that is resistant to outliers should produce consistent predictions regardless of support data. To implement this, we produce two prototypical-based classifiers from the two support sets of the two tasks. After that, we enforce the consistency of their predictions on query instances.

4.2 Model

4.2.1 Few Shot Learning for Event Detection.

In this work, the event detection problem is formulated as a $N + 1$-way K-shot episodic few-shot learning problem (V. D. Lai, Nguyen, & Dernoncourt, 2020; Vinyals et al., 2016). The model is given two sets of data: a support set $\mathcal{S}$ of

labeled data, and a query set $\mathcal{Q}$ of unlabeled data. $\mathcal{S}$ consists of $(N+1) \times K$ data points in which N is the number of positive event types and K is the number of samples per event type. The model is supposed to predict the labels of the data in the query set based on the observation of the novel event types given in the support set. Formally, a FSL task with a support set and a query set is defined as follows:

$$\begin{aligned} \mathcal{S} &= \{(s_i^j, a_i^j, y^j) | i \in [1, K]; j \in [0, N]\} \\ \mathcal{Q} &= \{(s_q^j, a_q^j, y_q^j) | q \in [1, Q]; j \in [0, N]\} \\ \mathcal{T} &= (\mathcal{S}, \mathcal{Q}); \quad \mathcal{Y} = \{y^j | j \in [0, N]\} \end{aligned} \tag{4.1}$$

where a data point (s_i^j, a_i^j, y^j) denotes a sentence s_i^j with trigger candidate a_i^j and event type y^j. Similar to prior studies in event detection, we add $y^0 = NULL$ to represent non-eventive type.

During training, development, and testing, the task $\mathcal{T}$ is sampled from three sets of data $\mathcal{D}^{train}$, $\mathcal{D}^{dev}$, and $\mathcal{D}^{test}$ whose sets of classes are $\mathcal{Y}^{train}$, $\mathcal{Y}^{dev}$, and $\mathcal{Y}^{test}$, respectively. These sets of classes are mutually disjoint to ensure that the model observes no more than K examples from a novel class.

A typical FSL model has two main modules: an encoder and a few-shot classifier. An encoder, denoted as ϕ, encodes an instance into a fixed-dimension vector

$$v_i^j = \phi(s_i^j, a_i^j) \in R^u \tag{4.2}$$

where u is the dimension of the representation vector. A few-shot classifier classifies a query instance among classes appearing in the support set. For instance, in a prototypical network, a prototype v^j is a class-representative instance that is an average of all vectors of the j-th class

$$v^j = \frac{1}{K} \sum_{i=1}^{K} \phi(s_i^j, a_i^j) \tag{4.3}$$

Then the distance distribution of the query instance $q = \{s_q, a_q, y_q\}$ (Snell et al., 2017) is:

$$P(q = y^j; \mathcal{S}) = \frac{e^{-d(v_q, v^j)}}{\sum_{k=1}^{N} e^{-d(v_q, v^k)}} \tag{4.4}$$

The training minimizes the cross-entropy loss, denoted by L_{ce}, over all query instances:

$$L_1(\mathcal{S}, \mathcal{Q}) = \sum_{q \in \mathcal{Q}} L_{ce}(y_q, P(q; \mathcal{S})) \tag{4.5}$$

4.2.2 Cross-task data augmentation.

In conventional episode training, two consecutive training tasks $\mathcal{T}_1$ and $\mathcal{T}_2$ are not likely to share an identical event type sets, $\mathcal{Y}_1 \neq \mathcal{Y}_2$. We assume that our training process has a memory to save the latest samples of every event type used in prior tasks. Using this memory, after a certain number of training iterations, for a new task $\mathcal{T}_1$, a second sample $\mathcal{T}_2$ can always be sampled from the memory such that $\mathcal{Y}_2 = \mathcal{Y}_1$. The expected value of delaying iterations for 5-way on the ACE dataset is 13 iterations ($stdev = 4$) and the RAM dataset is 98 iterations ($stdev = 24$) based on 1M simulations.

4.2.3 Prototype Across Task.

We are given two tasks $\mathcal{T}_1 = (\mathcal{S}_1, \mathcal{Q}_1)$ and $\mathcal{T}_2 = (\mathcal{S}_2, \mathcal{Q}_2)$ sampled with the same set of event type $\mathcal{Y}$. The prototypes are induced from both tasks as follows:

Let $E_1^S, E_2^S, E_1^Q, E_2^Q$ be the representation vectors of $\mathcal{S}_1, \mathcal{S}_2, \mathcal{Q}_1, \mathcal{Q}_2$, respectively, where $E_1^S, E_2^S \in R^{(N+1)K \times u}$ and $E_1^Q, E_2^Q \in R^{(N+1)Q \times u}$ (returned by ϕ). Then, an attention module, denoted by att, induces intermediate representations for the support and query instances of $\mathcal{T}_1$ via weighted sums of the support vectors of the $\mathcal{T}_2$, and vice versa:

$$\hat{H}_1^{(\cdot)} = att(E_1^{(\cdot)}, E_2^S) = \frac{1}{\sqrt{u}} \mathrm{sm}(E_1^{(\cdot)} (E_2^S)^T) E_2^S \tag{4.6}$$

$$\hat{H}_2^{(\cdot)} = att(E_2^{(\cdot)}, E_1^S) = \frac{1}{\sqrt{u}}\text{sm}(E_2^{(\cdot)}(E_1^S)^T)E_1^S \tag{4.7}$$

The final representations for both tasks are then the sum of their original representations and the cross-task representations:

$$H^{(\cdot)} = E^{(\cdot)} + \hat{H}^{(\cdot)} \tag{4.8}$$

Then, the prototypes for tasks $\mathcal{T}_1$ and $\mathcal{T}_2$ are computed by averaging vectors of the same class from $H_1^{\mathcal{S}}$ and $H_2^{\mathcal{S}}$, respectively (Snell et al., 2017).

4.2.4 Cross Task Consistency.

The Cross Task Consistency (CTC) further reduces the sample bias by introducing prediction consistency between classifiers generated from two tasks. Without loss of generation, we assume that one of the classifiers is impaired by poor sampling. We employ the knowledge distillation technique (Hinton, Vinyals, & Dean, 2015) that helps transfer knowledge from the stronger classifier to the weaker one. This thus makes the model more robust to the sample bias. We enforce the cross-task consistency by minimizing the differences between predicted label distributions from the classifiers of two tasks as follows:

$$L_2 = KL(f_{\mathcal{S}_1}(\mathcal{Q}_1), f_{\mathcal{S}_2}(\mathcal{Q}_1)) + KL(f_{\mathcal{S}_1}(\mathcal{Q}_2), f_{\mathcal{S}_2}(\mathcal{Q}_2)) \tag{4.9}$$

where $f_{\mathcal{S}}$ is a prototypical classifier trained from a support set $\mathcal{S}$ and KL denotes the Kullback–Leibler divergence.

Finally, to train the model, we minimize the total loss (α is a hyper-parameter):

$$L = L_1(\mathcal{S}_1, \mathcal{Q}_1) + L_1(\mathcal{S}_2, \mathcal{Q}_2) + \alpha L_2 \tag{4.10}$$

Testing: As the model does not have access to the prior task of the novel class, the prototypes are computed based on the vectors of the current task only. Hence, the model turns into the original Prototypical Network (Snell et al., 2017). Our

proposed methods only apply to the training process, hence, it provides a fair performance compared with prior FSL ED models.

4.3 Experiment

We evaluate the model on 5+1-way 5-shot and 10+1-way 10-shot FSL settings. As it has been observed that training with more classes helps improve the model performance, we use 18+1 classes during training, while keeping 5+1 and 10+1 novel classes during testing.

4.3.1 Dataset.

We evaluate the proposed model on three event detection datasets. **RAMS** is a recently released large scale dataset; it provides 9124 human-annotated event triggers for 139 event subtypes (Ebner et al., 2020). **ACE** is a benchmark dataset in event extraction with 33 event subtypes (Walker et al., 2006). **LR-KBP** is a large-scale event detection dataset for FSL. It merges ACE-2005 and TAC-KBP datasets and extends some event types by automatically collecting data from Freebase and Wikipedia (Deng et al., 2020). Since RAMS and ACE datasets are designed for supervised learning, we need to resplit them for FSL training. We use the exact training/development/testing split for ACE as presented in a prior study (V. D. Lai, Nguyen, & Dernoncourt, 2020). Following the same method, for RAMS, we merge the original training/development and testing splits. Then we discard 5 event subtypes [1] whose number of samples are not sufficient for sampling. Finally, we use event types: (*Artifact-Existence, Conflict, Contact, Disaster, Government, Inspection, Manufacture, Movement*) for training, (*Justice, Life*) for development, and (*Personnel, Transaction*) for testing. For the LR-KBP dataset, we follow the same 5-fold cross-validation procedure as (Deng et al., 2020), then report the

[1] conflict.attack.strangling, conflict.attack.hanging, contact.negotiate.n/a, movement.transportperson.fall, movement.transportperson.bringcarryunload

average performance. The numbers of event subtypes for the development and testing sets are set to 10 (Deng et al., 2020). The details of the splits are presented in Table 13.

Split	**RAMS**		**ACE-05**		**LR-KBP**[2]	
	#Classes	#Samples	#Classes	#Samples	#Classes	#Samples
Train	95	5,340	18	2,865	72	6,732
Dev	17	1,934	11	1,227	10	561
Test	22	1,793	11	1,226	10	1,291

Table 13. Statistics of three datasets: RAMS, ACE-05, and LR-KBP.

Encoder	**Model**	**RAMS**		**ACE-05**		**LR-KBP**	
		Dev	**Test**	**Dev**	**Test**	**Dev**	**Test**
5+1-way 5-shot							
BERTMLP	Proto	**79.7**	68.2	82.9	79.3	83.9	82.1
	InterIntra	**79.7**	69.2	82.7	79.8	84.9	82.4
	DMB-Proto	73.2	66.9	72.9	71.9	79.8	75.2
	ProAcT	**79.7**	**74.3**	**84.5**	**83.0**	**84.1**	**83.1**
BERTGCN	Proto	82.0	71.0	83.5	82.1	87.2	84.8
	InterIntra	81.3	72.4	82.8	82.3	87.1	85.0
	DMB-Proto	54.9	47.2	61.4	60.9	70.8	63.3
	ProAcT	**82.1**	**75.7**	**86.7**	**84.7**	**88.7**	**87.3**
10+1-way 5-shot							
BERTMLP	Proto	73.4	61.7	81.5	78.4	**80.7**	78.0
	InterIntra	**74.3**	61.8	81.4	78.5	80.2	78.4
	DMB-Proto	60.1	53.8	69.5	68.2	67.4	66.2
	ProAcT	73.2	**62.3**	**82.5**	**80.5**	**80.7**	**78.7**
BERTGCN	Proto	72.4	60.7	83.3	80.4	83.2	80.0
	InterIntra	**73.7**	61.9	83.0	80.7	82.8	80.5
	DMB-Proto	54.3	43.0	69.4	69.7	65.8	60.4
	ProAcT	73.6	**62.9**	**83.7**	**81.9**	**85.4**	**83.1**

Table 14. Performance (F-score) on the development and test sets of models on RAMS, ACE-05 and LR-KBP datasets on 5+1-way 5-shot and 10+1-way 10-shot settings

4.3.2 Baseline.

We consider three strong baselines for FSL ED. **Proto** features a prototype for each novel class and Euclidean distance function, presented in equation 4.4 (Snell et al., 2017). **InterIntra** is an extension of the prototypical network with two auxiliary training signals. It minimizes the distances among data points of the same class and maximizes the distances among prototypes (V. D. Lai, Nguyen, & Dernoncourt, 2020). **DMB-Proto** extends the prototypical network in a way that the representation vector for each data point is induced by a dynamic memory network running on the data of the same class (Deng et al., 2020). Since the source code of DMB-Proto is not published, we reimplement the few-shot classifier with a dynamic memory module (Xiong, Merity, & Socher, 2016). We examine two state-of-the-art BERT-based sentence encoders ϕ for ED, i.e. BERTMLP (S. Yang et al., 2019) and BERTGCN (V. D. Lai, Nguyen, & Nguyen, 2020b).

4.3.3 Hyperparameters.

In this work, stochastic gradient decent optimizer is used with learning rate $1e^{-4}$. The training/evaluation are set to 6,000 and 500 iterations respectively; the evaluation is done after every 500 training iterations. The dimension of the final representation is set to 512. We use a dropout rate of 0.5 to prevent overfitting. The coefficient of the cross-task consistency loss is set to $\alpha = 10$ based on the best development performance ($\alpha \in \{1, 10, 100, 1000\}$.

We evaluate our ED model using the micro F1-score. The training and evaluation are done on a single Nvidia GTX 2080Ti with 11GB of GPU RAM. The training and evaluation take approximately 4 hours. We implement the model using Pytorch version 1.6.0.

4.3.4 Result.

Table 14 reports the F-scores on the development and testing sets of the baselines and our proposed model (called **ProAcT**) on three datasets. There are two significant points from the table. First, using the same sentence encoders, ProAcT achieves the best performance on all three datasets and settings. The improvement margins are in range [1.0%-6.1%] on the 5-shot setting and [0.7%-3.1%] on the 10-shot setting. Second, the F-score margin between ProAcT and Proto decreases as the shot number increases. This indicates that the proposed model performs better when the number of observed samples is small. As the number of shots increases, the improvement gets saturated. This finding is parallel with the fact that sample bias is more likely when the number of shots is small. Hence, our proposed method is more suitable to event detection in few-shot learning schema, especially in the case where the number of shots is limited.

4.3.5 Ablation study.

Our proposed model involves three factors: the cross-task data (**data**), the cross-task attentive prototype (**attention**) and the cross-task consistency (**consistency**). To analyze the efficiency of these modules, we incrementally eliminate these modules from the full ProAcT model and evaluate the remaining model on 5+1-way 5-shot setting. If *attention* and *loss* are removed while *data* remains, the model and setting become a prototypical network with 5+1-way 10-shot setting during the training. This model has the same amount of support data that our model has during the training process. Note that the testing with novel classes remains 5+1-way 5-shot setting for every model. If the cross-task data is eliminated, the attentive prototype and consistency loss are also removed and the model and setting return to a prototypical network with 5+1-way 5-shot setting.

Model	Precision	Recall	Fscore
ProAcT (full model)	74.9	76.7	75.7
−attention	74.1	76.0	74.9
−consistency	73.3	75.7	74.4
−attention −consistency	72.5	74.5	73.4
−data (−attention −consistency)	69.9	72.4	71.0

Table 15. Ablation study of our proposed components on 5+1 ways 5-shot setting on the RAMS dataset with BERTGCN encoder.

Table 15 reports the performance on 5+1-way 5-shot FSL setting on RAMS with BERTGCN encoder. As shown in the table, removing any module leads to a decrease between [0.8%-1.3%] in performance. When both *attention* and *consistency* are eliminated, the performance drops of 2.3%. A further drop of 2.4% is seen if the cross-task data is eliminated. These suggest that the improvement originates from the use of cross-task data, the attention for prototype computation and the consistency of cross-task predictions.

4.3.6 Analysis.

To further analyze the efficiency of our proposed method, we aim to discover which classes benefit the most. To do that, we compute two confusion matrices for ProAcT and Proto models on the test set of RAMS. We fix the random seed to make sure the sampling during testing is identical between two runs, hence ensuring that the proportion of classes is identical. Figure 3 presents the difference between two confusion matrices exhibited by the proposed model ProAct and the prototypical network Proto. There are two major observations from the figure. First, overall ProAcT produces more accurate predictions than Proto, as shown on the diagonal. Second, ProAcT involves remarkably more correct predictions for negative examples than Proto. In the meantime, it generates a significantly lower number of errors in both false positive and false negative related to the NULL

	Other	personnel.elect.n/a	personnel.elect.winelection	personnel.endposition.firinglayoff	personnel.endposition.n/a	personnel.endposition.quitretire	personnel.startposition.hiring	personnel.startposition.n/a	transaction.transaction.embargosanction	transaction.transaction.giftgrantprovideaid	transaction.transaction.n/a	transaction.transfercontrol	transaction.transfermoney.borrowlend	transaction.transfermoney.embargosanction	transaction.transfermoney.giftgrantprovidea	transaction.transfermoney.n/a	transaction.transfermoney.payforservice	transaction.transfermoney.purchase	transaction.transferownership.borrowlend	transaction.transferownership.embargosanc	transaction.transferownership.giftgrantprov	transaction.transferownership.n/a	transaction.transferownership.purchase
Other	103	1	-3	-6	9	3	-10	-8	-26	-4	-18	-2	-5	-21	7	-15	-2	-11	3	-13	-2	-6	-11
personnel.elect.n/a	0	-7	-27	0	4	1	1	2	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0
personnel.elect.winelection	0	-14	0	0	1	0	-1	2	0	0	0	-1	0	0	0	0	0	0	0	0	0	1	0
personnel.endposition.firinglayoff	-9	0	0	-67	-19	-2	1	0	1	0	4	8	0	0	0	1	0	1	0	0	0	0	0
personnel.endposition.n/a	3	-1	0	-5	4	5	3	2	-1	-1	1	-5	0	0	0	-2	0	0	2	-1	0	-1	0
personnel.endposition.quitretire	-7	-2	0	3	11	21	-1	0	-1	0	0	-4	0	-2	0	0	0	0	0	0	0	0	0
personnel.startposition.hiring	-19	1	2	2	0	0	69	11	0	0	-1	0	-1	2	-1	-3	-3	-1	0	0	-4	-4	0
personnel.startposition.n/a	-15	2	0	2	3	0	21	23	0	0	-2	0	0	0	0	0	-1	0	0	0	-1	1	0
transaction.transaction.embargosanction	-9	0	0	0	-2	0	1	0	40	1	0	4	0	-8	-1	-1	0	-1	0	-24	0	0	0
transaction.transaction.giftgrantprovideaid	19	0	0	0	-1	1	-1	0	0	26	2	1	-1	0	29	-4	-2	1	-2	1	8	-1	2
transaction.transaction.n/a	7	0	0	0	2	-1	-1	-2	0	-6	14	-3	-1	-1	4	1	0	-10	1	0	-1	10	-2
transaction.transaction.transfercontrol	-6	0	0	-2	6	0	0	0	-2	0	-5	14	0	0	0	-3	-1	0	2	0	-1	-2	1
transaction.transfermoney.borrowlend	-2	0	0	0	0	0	1	1	0	1	-4	-2	-34	0	2	-3	-4	-3	12	0	4	3	-1
transaction.transfermoney.embargosanction	-11	0	0	0	-1	-1	0	-1	-2	3	-1	-3	0	-11	0	4	0	0	0	-17	1	-3	0
transaction.transfermoney.giftgrantprovideaid	3	0	0	0	0	0	-1	-1	0	29	8	1	-3	0	69	11	0	-2	5	2	-13	-6	-3
transaction.transfermoney.n/a	-10	0	0	-3	2	-2	-1	-1	-1	-1	-8	-2	-4	-2	-1	-36	-20	-6	-7	1	-6	-27	-8
transaction.transfermoney.payforservice	-30	0	0	-1	3	0	0	0	0	-5	4	-1	-8	-2	-5	-23	41	-2	3	1	1	4	1
transaction.transfermoney.purchase	-4	0	0	1	0	0	-3	1	2	2	-7	1	0	-1	-3	-5	1	18	-1	-1	0	-8	0
transaction.transferownership.borrowlend	-5	0	0	2	0	0	1	0	0	0	5	0	16	-1	6	13	4	2	5	0	-2	-8	-5
transaction.transferownership.embargosanction	-1	0	0	-2	1	-1	0	0	-21	0	-1	0	0	-11	0	-3	0	0	0	75	2	-1	0
transaction.transferownership.giftgrantprovideaid	11	0	0	0	0	0	0	0	0	21	-4	4	-7	0	-6	-5	1	1	1	-1	22	-9	-1
transaction.transferownership.n/a	-2	0	0	-1	2	0	-2	0	-1	2	9	-4	1	-1	4	-24	-9	0	-3	-3	1	-13	7
transaction.transferownership.purchase	-16	0	0	1	-1	0	-1	-1	0	0	12	2	-5	-1	0	1	-1	-8	3	0	-1	1	56

Figure 3. The differences of confusion matrices between ProAcT and Proto models. On the main diagonal, a positive value implies that ProAcT predicts more accurately than Proto, whereas, on the rest of the matrix, a negative value indicates that ProAcT creates less error than Proto. Visually, a green cell indicates that the prediction of ProAcT is more accurate than those from Proto. Red cells suggest the cases where Proto is better than ProAcT.

class, i.e. *Other* class in Figure 3, suggesting that our proposed model effectively mitigates the effect of noise introduced by the NULL class.

4.4 Related works

Prior studies in ED mainly follow the supervised learning scheme. The early work focuses on feature engineering with statistical models (Ahn, 2006; Hong et al., 2011; Ji & Grishman, 2008; Liao & Grishman, 2010). Recently, many deep learning architectures have been explored for automatic feature learning (Y. Chen et al., 2015; Feng et al., 2016; V. D. Lai, Nguyen, & Nguyen, 2020b; T. H. Nguyen, Cho, & Grishman, 2016; T. H. Nguyen & Grishman, 2015, 2018; Veyseh, Lai, et al., 2021). Some recent studies have also introduced methods to extending ED to new event types (Y. Chen et al., 2018; L. Huang et al., 2018; R. Huang & Riloff, 2012; V. D. Lai, Nguyen, & Dernoncourt, 2020; Liao & Grishman, 2011; T. H. Nguyen, Fu, et al., 2016; T. H. Nguyen et al., 2016g; Tong et al., 2020).

FSL has been extensively studied in computer vision (Fei, Lu, Xiang, & Huang, 2020; Finn et al., 2017; K. Lee et al., 2019; Snell et al., 2017; Vinyals et al., 2016). Recent work has also considered FSL for tasks in natural language processing (Bao et al., 2020; X. Han et al., 2018). For ED, prior FSL work has mostly relied on Prototypical network (Deng et al., 2020; V. D. Lai, Nguyen, & Dernoncourt, 2020). However, these models do not explore cross-task modeling as we do.

4.5 Summary

The contribution of this chapter includes:

– We propose to exploit the relationship between training tasks for few-shot learning event detection.

– We compute prototypes based on cross-task modeling and present a regularization to enforce the prediction consistency of classifiers across tasks.

– The experiment results show that exploiting cross-task relations can alleviate the poor sampling and outliers in the support set of the few-shot learning setting for ED.

In the last three chapters, we have proposed methods for event extraction with text written in English. While the world has more than 7,000 languages being used, there was little effort spent on studying EE methods for non-English languages. In the next two chapters, we present the first work in multilingual event-event relation extraction with a focus on event causality in chapter V and event hierarchy in chapter VI.

CHAPTER V

MULTILINGUAL EVENT CAUSALITY IDENTIFICATION

This chapter includes the materials from a published paper "*Viet Dac Lai, Amir Pouran Ben Veyseh, Minh Van Nguyen, Franck Dernoncourt, and Thien Huu Nguyen.* **MECI: A multilingual dataset for event causality identification.** In Proceedings of the 29th International Conference on Computational Linguistics*, pp. 2346-2356. 2022.*"

As the first author, Viet was responsible for the design of the annotation guideline, preprocessing the data for annotation, managing the annotation process, evaluation, and writing. Amir, Minh, Franck, and Thien gave meaningful intuition and a literature review of the event causality identification task. Amir provided the code base for the evaluation. Thien made the editorial revision of the submitted paper.

After exploring the learning method in chapter III for event detection, chapters V and VI switch the gear toward event-event relation extraction. Event-event relation extraction mainly concerns a few common relationships between two events such as causal, temporal, and subevent relations. In particular, this chapter will present the first work in multilingual even causality identification.

Event Causality Identification (ECI) is the task of detecting causal relations between events mentioned in the text. Although this task has been extensively studied for English materials, it is under-explored for many other languages. A major reason for this issue is the lack of multilingual datasets that provide consistent annotations for event causality relations in multiple non-English languages. To address this issue, we introduce a new multilingual dataset for ECI, called MECI. The dataset employs consistent annotation guidelines for five

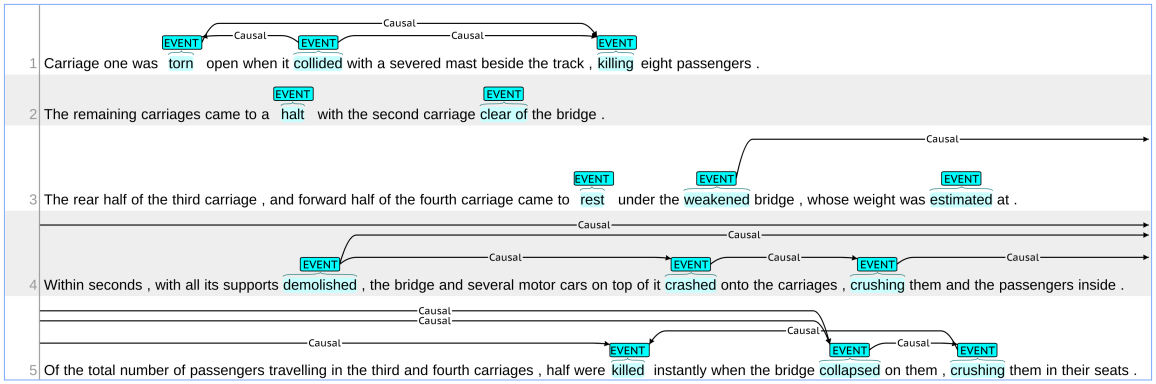

Figure 4. Our annotation interface for event causality identification.

typologically different languages, i.e., English, Danish, Spanish, Turkish, and Urdu. Our dataset thus enable a new research direction on cross-lingual transfer learning for ECI. Our extensive experiments demonstrate high quality for MECI that can provide ample research challenges and directions for future research.

5.1 Introduction

Event Causality Identification (ECI) is an important Information Extraction (IE) task that aims to identify causal relations between event mentions in text. For example, in the sentence *"After **inspection** of his computer , officers **found** that he was interested..."*, a ECI system should detect a causal relation between two events *"**inspection**"* $\xrightarrow{cause}$ *"**found**"*. ECI can provide valuable information for various applications such as event timeline construction (Shahaf & Guestrin, 2010), question-answering (Oh et al., 2016), future event forecasting (Hashimoto, 2019), and machine reading comprehension (Berant et al., 2014).

Due to its applications, ECI has been extensively studied in the natural language processing community over the past decade. The vast majority of methods for ECI involve feature engineering models (Do, Chan, & Roth, 2011; L. Gao, Choubey, & Huang, 2019; Hashimoto, 2019; Hu & Walker, 2017; Ning, Feng, Wu, & Roth, 2018) and recent deep learning architectures (Kadowaki, Iida,

Torisawa, Oh, & Kloetzer, 2019; J. Liu, Chen, & Zhao, 2021; Zuo et al., 2021a, 2021b). As such, the creation of large annotated datasets, e.g., EventStoryLine (Caselli & Vossen, 2017), has been critical to the development of the ECI study. However, existing datasets for ECI only annotate causal relations between event mentions in data of a single language, i.e., mainly for English (Caselli & Vossen, 2017; Cybulska & Vossen, 2014; O'Gorman, Wright-Bettner, & Palmer, 2016). On the one hand, this leaves many other languages unexplored for ECI, posing an important question about the generalization ability of existing methods to other languages. For instance, Spanish, Danish, and Turkish are not covered in those separate datasets for ECI. Moreover, the current single-language datasets for ECI tend to employ different annotation guidelines that prevent their combination into a larger corpus and cross-lingual transfer learning research to train and evaluate models in different languages. In all, the annotation discrepancy and limited language coverage hinder the research and development of the ECI in various dimensions, necessitating a new dataset with broader coverage for ECI.

To address this issue, this chapter introduces a Multilingual Event Causality Identification (MECI) dataset to standardize and foster future research in multilingual ECI. Particularly, we present a large-scale ECI dataset for five languages, i.e., English, Danish, Spanish, Turkish, and Urdu that are annotated with the same annotation guideline to enable cross-lingual transfer learning evaluation for the first time. As such, four languages, i.e., Danish, Spanish, Turkish, and Urdu, are not explored in any of the existing datasets for ECI. To facilitate open access to the dataset, we obtain the texts from Wikipedia for annotation in all examined languages. To make it consistent with prior research and benefit from the well-designed annotation guidelines of previous datasets, we inherit the event

schema from the ACE 2005 dataset (Walker et al., 2006), and the causal event relation guideline from EventStoryLine (Caselli & Vossen, 2017) (with both explicit and implicit causal relations) during the annotation process. In total, our MECI dataset involves 46K events and 11K relations that are substantially larger than those in existing ECI datasets.

In addition, we evaluate the proposed MECI dataset using state-of-the-art models for ECI. We investigate the challenges of MECI over all examined languages through the monolingual setting where the models are trained and evaluated in the same language. The experiments show that the performance of existing ECI models, even with large pre-trained language models (PLMs), is far from satisfactory; models for non-English languages generally perform poorer than their English counterparts. We also observe the importance of choosing language-specific or multilingual PLMs for ECI models as their effectiveness varies for different languages. Moreover, we evaluate the models in the zero-shot cross-lingual setting, where the models are trained on English data and tested on the data of the other languages. The experiment suggests transferability of ECI knowledge between English and Urdu while showing a significant performance drop in other language pairs. These results can serve as baselines for future studies on cross-lingual transfer learning for ECI. Finally, we report the analysis and challenges of the MECI dataset to provide insights for future ECI research. We will publicly release MECI to promote future studies in multilingual ECI.

5.2 Data Annotation

5.2.1 Annotation Scheme.

Our goal is to annotate causal relations between event mentions in text. To this end, we define the annotation scheme for event mentions following the

guidelines for the ACE 2005 dataset (Walker et al., 2006) for events, while the annotation guidelines for event causality relations are obtained from those for the EventStoryLine dataset (Caselli & Vossen, 2017). This allows us to inherit the well-designed documentation in such benchmark datasets and achieve consistency with prior research for ECI.

In particular, based on the ACE 2005 annotation guideline, an event in our dataset is either (1) an occurrence involving some participants, or (2) something that happens, or (3) a change of state. Event mentions/triggers are words/phrases in text that clearly evoke some event. As we are mainly interested in event causality relations, we only annotate event mention spans and do not include event types. To accommodate different languages, we allow event mentions/triggers to span multiple words in the sentences.

Next, for event causality relations, our annotation guideline follows the EventStoryLine dataset. In particular, a causal relation represents a directional relation between two events in which an event (CAUSE) causes another event (EFFECT) to happen or hold. This definition covers standard causal relations: cause, enablement, and prevention (Caselli & Vossen, 2017). In addition, similar to EventStoryLine, our dataset covers both explicit and implicit causality. Note that this is an extension from most prior annotation schema, i.e., Causal-TimeBank (Mirza & Tonelli, 2014), RED (O'Gorman et al., 2016), BECauSe (Dunietz, Levin, & Carbonell, 2017), that have only considered explicit relations covering the three causal concepts: *cause, enable,* and *prevent* through a verb-based lexicalization (Wolff, 2007). In our view, causality is a tool for humans to understand the world, and its existence is independent of the actual language for presentation (Neeleman & Van de Koot, 2012). Hence, event causality relations might be established

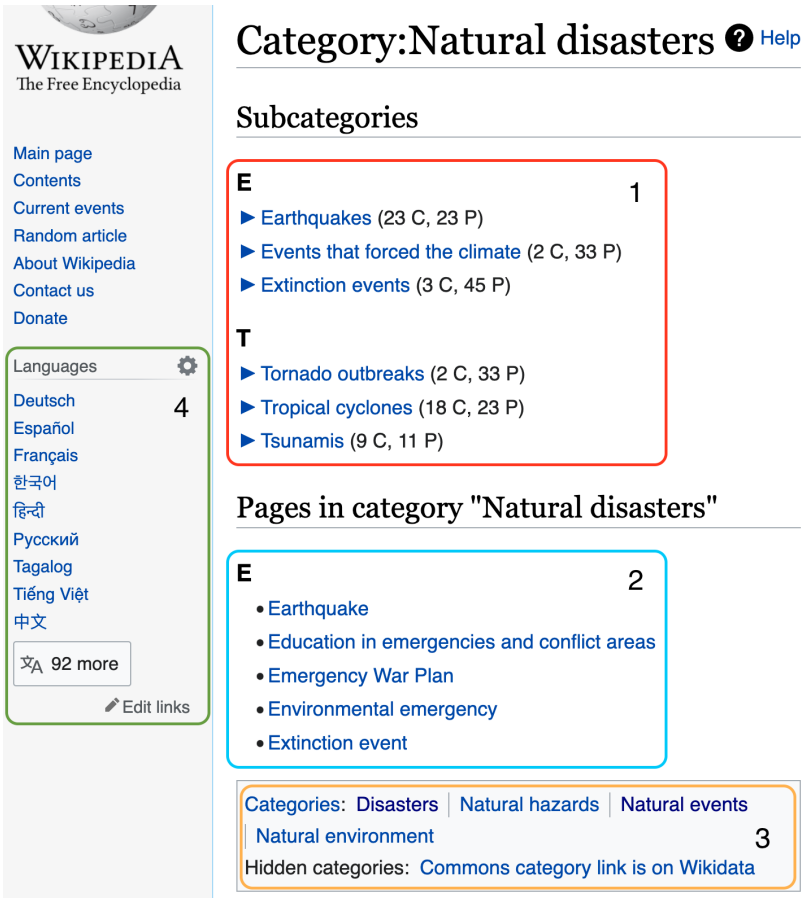

Figure 5. The Wikipedia category page of *Natural disasters* with its child categories (box 1, red), associated pages (box 2, cyan), parent categories (box 3, orange), and interlink to the same category in other languages (box 4, green).

without explicit ground in the text. In other words, there are implicit causal relations between events that are not covered by the above lexicalization (Caselli & Vossen, 2017; Webber, Prasad, Lee, & Joshi, 2019). To capture this important type of event causality relations, our annotation guideline is extended to cover implicit relations which require background knowledge, e.g., common-sense, domain-specific knowledge, for successful identification. Finally, similar to prior datasets, we annotate both intra- and inter-sentential causal relations between two events (Caselli & Vossen, 2017; Mirza & Tonelli, 2014).

5.2.2 Data Collection & Preparation.

The documents for our MECI dataset are collected from Wikipedia for five topologically different languages, i.e., English, Danish, Spanish, Turkish, and Urdu. In particular, we focus on 5 topics: aviation accidents, railway accidents, natural disasters, conflicts, and economic crisis, to expect a high yield of events and event causality relations. Wikipedia organizes articles into a hierarchical graph of categories. A category is a group of articles sharing a topic that might be further split into finer subcategories as shown in Figure 5. Furthermore, the hierarchical category systems in Wikipedia for different languages are interconnected through interlinks between identical categories. Therefore, by exploiting the category systems and language interlinks, we are able to obtain Wikipedia articles of the same topics across many languages.

Given the list of five categories for the examined languages, we crawl all the articles associated with their category descendants (i.e., subcategories, subsubcategories) in the hierarchy up to the depth of 6. After this step, we obtain at least 1,000 articles per category for each language. The obtained articles are cleaned by removing format elements (i.e., lists, images, URLs, and markups) to retain only textual data. Afterward, the articles are split into sentences and tokenized into words by Trankit (M. V. Nguyen, Lai, Pouran Ben Veyseh, & Nguyen, 2021), a multilingual text processing tool with state-of-the-art performance. The detailed list of subcategory URLs will be included in the final dataset package.

Given an article, a direct method for data annotation for ECI is to ask the annotators to label all the event mention spans and event mention pairs with causal relations. However, as the number of event mention pairs in a document grows quadratically with respect to the number of event mentions, a long Wikipedia

Language	Event	Relation
Danish	0.68	0.58
English	0.92	0.80
Spanish	0.84	0.66
Turkish	0.69	0.61
Urdu	0.65	0.75

Table 16. Kappa scores for the MECI dataset.

article can easily overwhelm the annotators, thus affecting the quality of the annotated data. To address the issue, we split the Wikipedia articles into smaller chunks that span five consecutive sentences for separate annotation, following prior practices (Ebner et al., 2020; Mostafazadeh, Grealish, Chambers, Allen, & Vanderwende, 2016). These chunks are called documents in our dataset. In this way, the annotators only need to consider a shorter context at a time to enhance the attention and quality of annotated data.

5.2.3 Human Annotation.

To annotate the obtained documents, we hire annotators from `upwork.com`, a crow-sourcing platform with freelancers from all around the globe. We only consider candidates that are (1) native to the target language, (2) fluent in English, and (3) highly approved among the Upwork employers. We can access this information from the annotators' profiles on the platform. The candidates are then given annotation guidelines and a test for performing both event annotation and event causality relation extraction tasks. The top two candidates are hired for each language. We use the BRAT annotation tool for our annotation (Stenetorp et al., 2012) and illustrated in Figure 4.

Our annotation consists of two tasks, i.e., event mention annotation and event causal relation annotation. For each language, we annotate event causality relations over the outputs from event mention annotation (i.e., after event mention

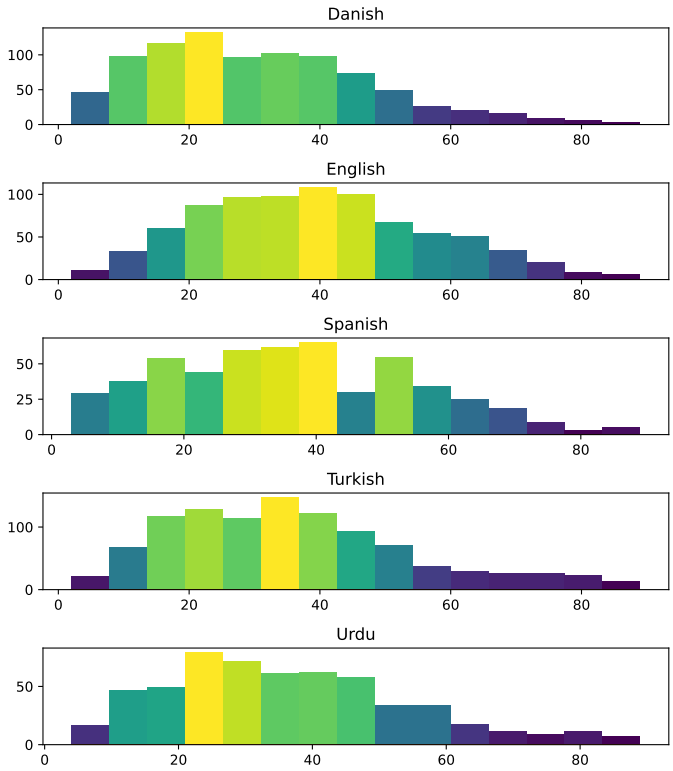

Figure 6. Distributions of distances between two event mentions with causal relations in MECI. Distances are measured via the number of words.

annotation has been completed and finalized for all documents). Given a sample of selected documents for a language, for each task, the two annotators for that language independently annotate event mentions/event causal relations for the documents. Afterward, the annotation conflicts will be presented to the annotators for further discussion and revision to produce the final version of the annotated documents for the current task. This will help to ensure high agreement and consistency for our dataset.

5.2.4 Data Analysis.

Table 16 presents our Kappa scores for annotation agreements of event mentions and event causality relations over different languages. Note that these scores are computed by comparing the independent annotations of the annotators

over the documents before engaging in discussion to resolve conflicts. As can be seen, the scores are very close to either substantial or almost perfect agreement for all the tasks and languages, thus demonstrating the high quality of our created MECI dataset. We also find that non-English languages tend to have lower annotation agreement scores for both event mention and causality relation extraction tasks, thus highlighting the challenges of ECI for non-English languages and showing the importance of additional research for multilingual ECI.

In addition, Table 17 show other statistics for our MECI dataset. Across five languages, each document contains an average of 13.0 event triggers, which account for 2.6 event triggers per sentence. This reveals a challenge of MECI for ECI models that might need to handle the ambiguity due to the overlap of the context of event mention pairs in both sentence and document levels. Furthermore, each document contains approximately 3.1 relations on average; however, there is a discrepancy in event causality relation density in documents among languages. In particular, English and Turkish represent a much denser level of event causality relations per document than other languages, especially Spanish and Urdu. As such, the divergences in the density of event causality relations (and event mentions) pose another robustness challenge for ECI models that should be able to bridge the gaps and transfer event causal knowledge across languages.

Finally, Figure 6 presents the distributions of distances between two event mentions with causal relations for five examined languages in MECI (the distances are counted via the number of words in between). There are several observations from the figure. First, for all the languages, a majority of event mentions are 10 to 50 words away from each other in the documents. This suggests diverse levels of context information between event mentions that an ECI model needs to capture

to perform well for the languages in MECI. Second, there are clear divergences between the distance distributions of causal event mention pairs over languages. For instance, the distances between event mentions for Danish and Urdu seem to be more distributed in the shorter ranges than those of English and Spanish. Such distribution differences require ECI models to introduce robust mechanisms to induce language-transferable representations for diverse causal contexts in cross-lingual learning for ECI.

5.2.5 Dataset Comparison.

Table 17 also compares our MECI dataset with previous public datasets for ECI such as Causal-TimeBank (Mirza, Sprugnoli, Tonelli, & Speranza, 2014), RED (O'Gorman et al., 2016), BECauSE-2.0 (Dunietz et al., 2017) , CaTeRS (Mostafazadeh et al., 2016), and EventStoryLine (Caselli & Vossen, 2017). We also include some monolingual ECI datasets for Arabic and Persian such as SACB Sadek and Meziane (2018) and PerCause Rahimi and Shamsfard (2021). Note that we focus on the datasets that explicitly consider causal relations between event mentions/triggers to make them comparable. It is clear from the table that our MECI dataset has a much larger scale with more event mentions, causal relations, and languages than all previous datasets for ECI. This will enable the training of larger models and a more comprehensive evaluation for ECI.

5.2.6 Challenges.

Unlike most prior ECI datasets, our MECI dataset includes implicit causal relations, which allow causal relations to be derived from various implicit reasoning sources such as common-sense knowledge. This section illustrates some types of implicit reasoning for causal relations between events discovered in our dataset.

Dataset	Language	#Docs	#Rels	#Events	Relation Type
Causal-TimeBank	English	100	318	11,000	Explicit
BECauSE-1.0		1200	400	-	Explicit
RED		95	*4,969	8,731	Explicit
BECauSE-2.0		118	1,803	-	Explicit
CaTeRS		320	488	2,708	Explicit, Implicit
EventStoryline		258	5,519	7,275	Explicit, Implicit
SACB	Arabic	-	2,162	-	-
PerCause	Persian	-	5,128	-	-
MECI	Danish	519	1,377	6,909	Explicit, Implicit
	English	438	2,050	8,732	
	Spanish	746	1,312	11,839	
	Turkish	1,357	5,337	14,179	
	Urdu	531	979	4,975	
MECI (total)	Various	3591	11,055	46,634	Explicit, Implicit

Table 17. Comparison of public ECI datasets. #Relations indicates the number of causal relations in the datasets. * designates the numbers that include other event-event relations, i.e., temporal and hierarchical relations.

Implicit inference of causal cues: In the following example, considering two event mentions: *"derailed"* and *"running into"*, there is no triggering verb-based expression to signal the causal relationship between the two events. However, with the presence of the trailing comma between the two event mentions, our annotators can easily realize that the *"derail"* event is the cause of the *"running into"* event. As such, the annotators might have implicitly inferred the reduced relative clause "*which makes the train*" (presented in the brackets) between the two event mentions to make the causal decision. To this end, a model will also need to recognize such implicit reasoning cues based on the context to successfully perform ECI.

> *The Granville rail disaster ... when a crowded commuter train* ***derailed****,* *[~~which makes the train~~]* ***running into*** *the supports of a road bridge that ...*

Implicit transitivity: Consider three event mentions *"trouble"*, *"bail out"*, and *"killed"* in the following example. The ground text explicitly expresses the causal relation *"bail out"* $\xrightarrow{cause}$ *"killed"* via the adverb *"consequently"*. However, there is no clear signal of the causality between *"trouble"* and *"bail out"*, which requires common-sense knowledge to successfully recognize for the causal order of such events, i.e., *"trouble"* $\xrightarrow{cause}$ *"bail out"*. This increases the difficulty for identifying the causality *"trouble"* $\xrightarrow{cause}$ *"killed"*, which might entail transitivity reasoning between implicit and/or explicit causal relations, i.e., *"trouble"* $\xrightarrow{cause}$ *"bail out"* and *"bail out"* $\xrightarrow{cause}$ *"killed"*.

> *... when his Spitfire developed engine* ***trouble*** *between the islands of Skiathos and Skópelos over the Aegean Sea . He attempted to* ***bail out*** *of the aircraft, but his altitude was too low for his parachute to open, and he was consequently* ***killed****.*

5.3 Experiments

We randomly split the documents for each language in MECI into three separate parts with a ratio of 3/1/1 to serve as training, development, and test data respectively for experiments. To study the challenges of ECI presented in MECI, we evaluate the performance of the state-of-the-art models for ECI on this dataset. Each model will be comprehensively evaluated in the monolingual learning (i.e., trained and tested on data of the same language) and multilingual learning (i.e., trained and tested on the data of different language) settings with MECI.

5.3.1 ECI Models.

We explore the following representative models for ECI in the literature:

PLM: This model is inherited from the BERT baseline in (Tran Phu & Nguyen, 2021). Given an input document D, this model concatenates the words

	Model	MECI English			EventStoryLine		
		Precision	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**
BERT	PLM	35.6	44.9	39.7	27.3	35.3	30.8
	RichGCN	**48.1**	**69.5**	**56.8**	**42.6**	**51.3**	**46.6**

Table 18. Performance of models on MECI (English) and EventStoryLine datasets.

from all sentences and sends it into a pre-trained language model, e.g., BERT (Devlin et al., 2019), to obtain representation vectors for each word-piece using the hidden vectors in the last transformer layer. Afterward, given the spans A and B for two event mentions e_A and e_B of interest in D, we compute the representations $\mathbf{r}_A$, $\mathbf{r}_B$ for the two event mentions by averaging the representation vectors of the word pieces within the corresponding spans A and B. Finally, we form an overall representation vector $\mathbf{r}_{A \to B} = [\mathbf{r}_A, \mathbf{r}_B, \mathbf{r}_A - \mathbf{r}_B, \mathbf{r}_A * \mathbf{r}_B]$ ($*$ is the element-wise multiplication operation) for ECI. This vector will be fed into a feed-forward network with a sigmoid function in the end to predict the causal relationship between e_A and e_B in D.

RichGCN (Tran Phu & Nguyen, 2021): Similar to **PLM**, **RichGCN** employs a PLM to encode the entire input document and compute an overall representation vector $\mathbf{r}_{A \to B}$ for identifying the causal relationship between two given event mentions. To enhance representation learning, **RichGCN** also introduce several interaction graphs (with words and event mentions in the input document as the nodes) to capture relevant context information/interactions for the causal relationship between two event mentions. In particular, to adapt **RichGCN** to MECI with multiple languages, we implement four interaction graphs to represent an input document: (1) *Sentence Boundary Graph* where words or event mentions within each sentence in the document are connected to each other; (2) *Event Mention Span Graph* where words within each event mention span

are connected to the event mention; (3) *Syntax-based Graph* where words within each sentence are connected to each other following the dependency tree structure of the sentence; and (4) *Semantic-based Graph* where words across the document are connected to each other; the weights for the connections are measured via the similarity between the word representations (computed from PLM). In **RichGCN**, each interaction graph is represented by an adjacency matrix. A final graph V to capture relevant connections for the two event mentions is formed by learning a linear combination of the adjacency matrices of the four graphs. Finally, the graph V is then sent into a Graph Convolutional Network (GCN) (Kipf & Welling, 2017) to compute a richer representation for the two event mentions with more relevant context to perform ECI.

Know (J. Liu et al., 2021): By treating the event mentions as concepts in ConceptNet (Speer, Chin, & Havasi, 2017), **Know** retrieves related concepts and relations for the two input event mentions in our ECI problem from ConceptNet. The retrieved information is then used to augment the input text. As such, **Know** also utilizes a PLM to encode the augmented text to compute prediction representation for ECI. In addition, this model employs a masking mechanism to obtain event-agnostic context from input text, serving as another source of information to be encoded by the PLM and incorporated into representation learning for our task.

5.3.2 Experiment Setups.

In the monolingual learning settings, for each language in MECI, we train the ECI models on the training data and evaluate model performance on the test data of the same language. We explore both multilingual PLMs, i.e., mBERT (Devlin et al., 2019) and XLMR (Conneau et al., 2020), and language-

specific PLMs for the languages in MECI as the encoder for the ECI models in the experiments. In particular, we utilize the following language-specific PLMs that are available for MECI languages, i.e., BERT (Devlin et al., 2019) for English; BotXO[1] for Danish, BETO (Cañete et al., 2020) for Spanish, BERTurk (Schweter, 2020) for Turkish, and UrduHack[2] for Urdu.

The support of multiple languages with the same annotation guideline for event causality relations in MECI allows us to perform cross-lingual transfer learning evaluation for ECI models. In particular, for cross-lingual settings, ECI models are trained on the training data of one language (the source language); however, they are evaluated on test data of new target languages. In the experiments, we treat English as the source language and other languages in MECI as the target languages for cross-lingual evaluation. To facilitate the prediction over multiple languages, we leverage the multilingual PLMs mBERT and XLMR in cross-lingual experiments.

Hyper-parameters: We employ the same hyper-parameters from the original works for the ECI models: **RichGCN** (Tran Phu & Nguyen, 2021), and **Know** (J. Liu et al., 2021) in the experiments. The multilingual NLP toolkit Trankit (M. V. Nguyen, Lai, Pouran Ben Veyseh, & Nguyen, 2021) is leveraged to obtain dependency trees for sentences in multiple languages for the **RichGCN** model. Also, we utilize the multilingual version of ConceptNet (Speer et al., 2017) to retrieve augmented information for **Know**. Finally, we employ the base versions for all the multilingual and monolingual PLMs considered in this work.

[1]https://huggingface.co/Maltehb/danish-bert-botxo

[2]https://github.com/urduhack/urduhack

	Model	English			Danish			Spanish		
		P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
mBERT	PLM	38.4	46.0	41.9	25.2	26.6	25.9	43.9	41.5	42.7
	Know	35.8	56.7	43.9	25.8	36.0	30.1	39.7	38.3	39.0
	RichGCN	48.4	67.1	56.2	29.7	38.0	33.4	**51.2**	52.0	51.6
XLMR	PLM	48.7	59.9	53.7	**35.9**	36.2	36.0	50.6	49.1	49.9
	Know	39.3	42.6	40.9	31.4	11.4	16.7	39.9	28.4	33.2
	RichGCN	**50.6**	**68.0**	**58.1**	31.9	**50.0**	**38.9**	50.7	**55.0**	**52.8**

	Model	Turkish			Urdu		
		P	**R**	**F**	**P**	**R**	**F**
mBERT	PLM	36.2	48.7	41.6	31.9	34.3	33.0
	Know	39.7	46.9	43.0	36.7	35.3	36.0
	RichGCN	50.0	59.9	54.5	40.1	50.0	44.5
XLMR	PLM	44.0	59.4	50.5	40.4	43.2	41.8
	Know	36.5	46.7	41.0	**41.1**	22.2	28.9
	RichGCN	**50.5**	**64.6**	**56.7**	37.7	**56.0**	**45.1**

Table 19. Monolingual learning performance of ECI models on MECI with mBERT and XLMR.

5.3.3 Monolingual Performance.

Table 19 shows the performance of the three ECI models on the monolingual learning settings across all the languages with the multilingual PLMs: mBERT and XLMR. Among the ECI models, we find that **RichGCN** maintains its top performance across all the languages and multilingual PLMs, thus demonstrating the effectiveness of its language-agnostic document structure to represent documents for ECI. Nonetheless, the best performance by **RichGCN** for English, Danish, Spanish, Turkish, and Urdu is 58.1, 38.9, 52.8, 56.7, and 45.1. This performance is far from being perfect, thus suggesting the challenges for ECI across languages and presenting ample research opportunities to improve the performance in the future. In addition, among the models, **Know** exhibits mixed performance with mBERT and worst performance with XLMR across languages. We attribute this phenomenon to the unstable quality of the concept retrieval with ConceptNet

and context modification in **Know** that might exclude important causal context from the input texts to cause poor performance in different languages. Finally, comparing the multilingual PLMs, we find that XLMR performs significantly better than mBERT over all the languages with the **PLM** and **RichGCN** models, thus suggesting the benefits of XLMR for future ECI research.

5.3.4 Effects of language-specific PLMs.

To better understand the effectiveness of PLMs for ECI, Table 20 reports the performance of **PLM** and **RichGCN** in the monolingual learning settings where language-specific PLMs for each language are employed as the encoder for the models. As can be seen, using the best model **RichGCN** and the best multilingual PLM XLMR as the anchors, ECI performance for English, Spanish and Turkish is very close with monolingual and multilingual PLMs (i.e., less than 2% difference in F1 scores). However, multilingual PLMs are substantially better than monolingual PLMs for Danish and Urdu (up to 7% difference in performance). This can be attributed to the lower resources in Danish and Urdu that hinder effective training for language-specific PLMs. With multilingual PLMs, such low-resource languages can benefit more from data in other languages to train multilingual PLMs.

Language	**PLM**			**RichGCN**		
	P	**R**	**F**	**P**	**R**	**F**
English	35.6	44.9	39.7	**48.1**	**69.5**	**56.8**
Danish	23.2	23.0	23.1	**27.1**	**35.0**	**30.6**
Spanish	42.7	44.6	43.6	**59.8**	**48.2**	**53.4**
Turkish	40.4	56.0	46.9	**54.7**	**62.0**	**58.1**
Urdu	20.2	33.5	25.2	**31.1**	**47.9**	**37.7**

Table 20. Monolingual learning performance of ECI models on MECI with language-specific PLMs.

Embedding	Model	P	R	F	P	R	F
		English → Danish			**English → Spanish**		
mBERT	PLM	12.4	35.4	18.4	11.4	63.3	19.3
	Know	7.8	**62.0**	13.8	7.2	**69.4**	13.0
	RichGCN	23.7	45.3	31.1	20.6	58.6	30.5
XLMR	PLM	20.1	59.2	30.1	16.0	66.4	25.8
	Know	13.3	42.1	20.3	10.4	47.3	17.1
	RichGCN	**28.5**	43.7	**34.5**	**22.7**	62.4	**33.3**
		English → Turkish			**English → Urdu**		
mBERT	PLM	21.5	47.6	29.6	17.0	44.2	24.6
	Know	20.4	55.5	29.9	14.2	61.5	23.0
	RichGCN	44.5	52.0	48.0	35.0	56.8	43.3
XLMR	PLM	36.1	**60.5**	45.2	25.7	**62.0**	36.3
	Know	25.8	57.6	35.7	19.3	54.5	28.5
	RichGCN	**46.4**	55.0	**50.3**	**38.6**	55.2	**45.5**

Table 21. Zero-shot cross-lingual learning performance on MECI using English as source language.

5.3.5 Cross-lingual Performance.

To investigate the transferability of ECI knowledge across languages, Table 21 presents the performance of the ECI models in the cross-lingual learning settings. Note that in these experiments English is the source languages while other languages are the targets. Among the three models, **RichGCN** is still the best performer across all target languages. However, the model's performance drops significantly for the three target languages Danish (by 4.4%), Spanish (by 19.5%), and Turkish (by 6.4%) compared to their monolingual performance with XLMR. This illustrates the challenges and necessity of further research on cross-lingual transfer learning for ECI that can now be enabled with our multilingual dataset.

Interestingly, compared to the monolingual settings, the performance on Urdu of **RichGCN** is slightly improved (by 0.4%) in the cross-lingual setting. One potential reason is due to the smallest size of the training data for Urdu in MECI that allows the larger English training data to train better models for Urdu test

data. In addition, among the four target languages, we observe a wide range of cross-lingual performance from the model trained on English data, thus showing the diverse nature of data and languages in MECI for future research.

5.4 Related Work

As an important task in IE, ECI has attracted extensive research effort to develop effective models (Do et al., 2011; Hashimoto et al., 2014; Hidey & McKeown, 2016; Hu & Walker, 2017; Kadowaki et al., 2019; J. Liu et al., 2021; Tran Phu & Nguyen, 2021; Zuo, Chen, Liu, & Zhao, 2020). To support model development for ECI, several datasets have been introduced for this task, including PDTB (Prasad et al., 2008), Causal-TimeBank (Mirza, 2014), ECB (Cybulska & Vossen, 2014), Richer Event Description (O'Gorman et al., 2016), BeCause (Dunietz et al., 2017), and EventStoryLine (Caselli & Vossen, 2017), CaTeRS (Mostafazadeh et al., 2016). However, these previous works and datasets only focus on English data, presenting a strong demand for new research and datasets on other languages for ECI.

To this end, there are a few efforts on creating causality corpora for other languages, such as German (Rehbein & Ruppenhofer, 2020), Arabic (Sadek & Meziane, 2018) and Persian (Rahimi & Shamsfard, 2021). However, these corpora consider not only event mentions, but also entities, clauses, and sentences, thus, not directly solving ECI as we do. In addition, most existing annotation efforts for ECI focus on explicit event causality relationships. EventStoryLine (Caselli & Vossen, 2017) and CaTeRS (Mostafazadeh et al., 2016) are the only two prior datasets that also explore implicit causal relationships between events. However, they do not provide annotation for multiple languages as we do in MECI.

5.5 Summary

The contribution of this chapter includes:

- We present a new dataset for event causality identification in five different languages across diverse typologies. The dataset is annotated consistently for all languages, offering a large number of event mentions/causal relations and covering four languages that have not been explored in the prior ECI resources.
- Our extensive experiments and analysis reveal the quality and challenges of our dataset for the multilingual ECI task.
- In addition, our dataset enables cross-lingual transfer learning research that is not possible with current resources for ECI.

While this chapter has presented the first work for multilingual event causality identification, there are other types of event-event relations such as event hierarchy (subevent relation) and event co-reference. The next chapter investigates the first work in multilingual subevent extraction with the creation of a subevent extraction corpus and a language agnostic to select a better context for event-event relation extraction.

CHAPTER VI

MULTILINGUAL SUBEVENT RELATION EXTRACTION

This chapter includes the materials from a published paper "*Lai, Viet, Hieu Man, Linh Ngo, Franck Dernoncourt, and Thien Nguyen. "Multilingual SubEvent Relation Extraction: A Novel Dataset and Structure Induction Method." In Findings of the Association for Computational Linguistics: EMNLP 2022, pp. 5559-5570. 2022.*

As the first author, Viet was responsible for the design of the annotation guideline, preprocessing the data for annotation, managing the annotation process, evaluation, and writing. Hieu was responsible for the development of the OT model, and Linh and Thien gave meaningful discussions and insights. Thien made the editorial revision of the submitted paper.

Continue the work of multilingual event-event relation extraction in chapter V, this chapter presents a similar work for multilingual subevent relation extraction. Subevent Relation Extraction (SRE) is a task in Information Extraction that aims to recognize spatial and temporal containment relations between event mentions in text. Recent methods have utilized pre-trained language models to represent input texts for SRE. However, a key issue in existing SRE methods is the employment of sequential order of words in texts to feed into representation learning methods, thus unable to explicitly focus on important context words and their interactions to enhance representations. In this work, we introduce a new method for SRE that learns to induce effective graph structures for input texts to boost representation learning. Our method features a word alignment framework with dependency paths and optimal transport to identify important context words to form effective graph structures for SRE. In addition, to enable SRE research on non-English languages,

we present a new multilingual SRE dataset for five typologically different languages. Extensive experiments reveal the state-of-the-art performance of our method on different datasets and languages.

6.1 Introduction

In Information Extraction (IE), events are defined as things that happen/occur (Pustejovsky, Castaño, et al., 2003) or changes of state of real-world entities (Walker et al., 2006). Due to their complexity, a general event (i.e., superevent) can involve multiple other events with finer granularity (i.e., subevents) that can be altogether mentioned in text to present necessary details (e.g., a *war* can contain multiple *attacks*, which, in turn, can contain different bombing events). This work studies the problem of subevent relation extraction (SRE): given two event mentions in a document, a model needs to predict if one even is a part/subsevent of the other one. Following previous work (Glavaš, Šnajder, Moens, & Kordjamshidi, 2014), our SRE problem requires that a subevent relation is only established if the subevent is both spatially and temporally contained in the superevent. Accordingly, SRE systems will need to effectively model document context to infer spatiotemporal evidences for subevent reasoning. Among others, SRE finds its important applications in summarization (Filatova & Hatzivassiloglou, 2004) and information retrieval (Glavaš & Šnajder, 2013).

To encode document context, existing models (Trong, Ngo, Ngo, & Nguyen, 2022; H. Wang, Chen, Zhang, & Roth, 2020) have leveraged pre-trained language models, i.e., RoBERTa (Y. Liu et al., 2019), to obtain representations for input documents for subevent prediction. However, an issue of existing SRE methods is that they only rely on the sequential format of documents (i.e., sequence of sentences/words) for representation learning. On the one hand, the sequential

format does not provide mechanisms to highlight the most important context words or avoid irrelevant ones in input documents, potentially introducing noisy information in the representations for SRE. Further, due to the sequential nature of input texts, current SRE models cannot exploit effective structures/graphs that directly connect important context words to improve representation learning for SRE.

Motivated by recent works on relation extraction between entities (Gupta, Rajaram, Schütze, & Runkler, 2019; Sahu, Christopoulou, Miwa, & Ananiadou, 2019; Y. Zhang et al., 2018), one approach to improve the sequential representation of input texts for SRE can be based on dependency trees of sentences (i.e., graph-based structures) where dependency paths (DP) between two input entity mentions have been shown to capture important context words. In particular, to adapt this idea to the document level with multiple sentences, (Gupta et al., 2019) obtains dependency trees for each sentence whose roots are linked together to obtain connected dependency graphs for input documents. Afterward, the dependency graphs for documents are pruned to preserve only the words along the dependency paths between two input mentions (called in-DP words) for representation learning. However, for our SRE problem, important context words for subevent prediction can also be distributed outside the dependency paths, thus necessitating further techniques to identify other important words and connect them with the in-DP words to form better graph structures to represent input texts for SRE.

> *"They* **implemented** *the proposal early last year. Following the plan, the performers* **collected** *data and* **developed** *frameworks to monitor human trafficking for the first step of the proposal."*

For example, in the above text, "*developed*" is a subevent of the "*implemented*" event for which the DP is "*implemented* $\rightarrow$ *collected* $\rightarrow$ *developed*". However, the word "*proposal*", which is important to connect "*implemented*" and "*developed*" to the same target for subevent recognition, is not included in the DP in this case. For convenience, we use non-DP words to refer to the words that do not belong to the DPs between two input event mentions for SRE.

In previous work, in-DP words can be extended to find additional important context words for relation prediction by including non-DP words close to the DPs in the dependency graphs (Y. Zhang et al., 2018) (i.e., based on syntactic distances). As such, this method does not consider contextual semantics of the words that can provide richer information for important word selection for SRE. To address this issue, we propose to leverage both syntactic and semantic evidences to determine the importance of a non-DP word for inclusion into the graph structure to represent input text for SRE. For syntactic information, we expect a word to be more important for subevent prediction if it is closer to the input event mentions in the dependency graphs. In addition, for semantic information, our intuition is to promote non-DP words that are more similar/related to in-DP words contextually to enhance the induced representations for SRE. However, combining syntactic and semantic similarities to compute overall importance scores to compare non-DP words is a non-trivial problem due to the different nature of the information. To this end, motivated by in-DP words as the anchors to induce graph structure representations for input texts, we propose to cast the problem of combining syntactic and semantic similarities to select important non-DP words into finding an optimal alignment between non-DP and in-DP words. A non-DP word is considered to be important for SRE and retained in the induced graph

structures for input texts if it is aligned with one of the in-DP words. In this way, our approach facilitates the application of Optimal Transport (OT) methods to effectively integrate syntactic and semantic information into a single joint optimization problem to obtain the optimal alignment for non-DP word selection for SRE. In particular, to adapt to the goal of aligning two groups of points based on their transportation costs and distributions in OT, we will leverage semantic similarity to obtain transportation costs while syntactic distances in dependency graphs will be used to compute the distributions for in-DP and non-DP words to perform word alignment for SRE. The resulting word alignment will then be used to select important non-DP words and construct graph structures to learn representations for subevent prediction.

We evaluate our method over HiEve (Glavaš et al., 2014) and Intelligence Community (IC) (Hovy, Mitamura, Verdejo, Araki, & Philpot, 2013), popular public datasets for SRE. However, an issue with prior datasets and methods for SRE is that they are only developed and evaluated over English data. As such, a critical question for the generalization of SRE methods to non-Enlgish languages has not been explored in the literature. To address this issue, we further present a new multilingual dataset for SRE (called mSubEvent) for five languages, i.e., English, Danish, Spanish, Turkish, and Urdu, to enable future research in multilingual learning for SRE. Our dataset follows the annotation guidelines in HiEve to make it consistent with prior SRE work, introducing a large SRE dataset with more than 46K event mentions and 3.9K subevent relations for model development. We conduct extensive experiments over HiEve and our new dataset mSubEvent to demonstrate the effectiveness of the proposed method with state-of-the-art performance for SRE. Our experiments cover both monolingual learning

(i.e., training and test data are from the same language) and cross-lingual transfer learning evaluation (i.e., training and test data comes from different languages), thus highlighting the generalization across languages of the proposed method for SRE. To our knowledge, this is the first work that explores multilingual data and cross-lingual learning for SRE. Finally, we will publicly release the new mSubEvent dataset to provide baselines and resources for future research in this area.

6.2 Data Annotation

There exist several datasets with subevent relation annotation, including HiEve (Glavaš et al., 2014), IC (Araki, Liu, Hovy, & Mitamura, 2014; Hovy et al., 2013), and RED (O'Gorman et al., 2016). However, these datasets are only annotated for English data, thus unable to evaluate the generalization of models across multiple languages. To better evaluate the proposed model and enable future research on multilingual SRE, we introduce the first multilingual dataset (called mSubEvent) for SRE that provides human annotation for five typological different languages, i.e., English, Danish, Spanish, Turkish, and Urdu. The rest of this sections describes our annotation schema, data collection, and annotation efforts.

Annotation Scheme: A dataset for SRE needs to provide annotations for two tasks, i.e., event mention and subevent relation extraction. As such, we inherit the well-designed annotation guidelines from existing benchmark datasets for both tasks to be consistent with prior work. In particular, we employ the annotation guideline and definition for event mentions from the popular ACE-2005 dataset (Walker et al., 2006). As our dataset focuses on subevent relations, we only annotate event mention spans and do not provide event types to reduce annotation cost. We allow event mentions to span multiple consecutive words in a sentence to

Language	Event	Relation
English	0.92	0.96
Danish	0.68	0.83
Spanish	0.84	0.78
Turkish	0.69	0.66
Urdu	0.65	0.88
Average	0.75	0.82

Table 22. Kappa agreement scores.

flexibly handle different languages. In addition, for subevent relation annotation, we follow the guidelines from HiEve (Glavaš et al., 2014), a popular dataset for SRE. Following recent work (H. Wang et al., 2020), our dataset assigns a relation label for each pair of annotated event mentions in a document using three labels, i.e., `PARENT-CHILD`, `CHILD-PARENT`, and `NOREL`.

Data Collection & Preparation: To enable public release of our dataset, we collect documents for annotation from Wikipeda of the five intended languages. In particular, we obtains document from five event-intensive topics/categories in Wikipedia, including aviation accidents, railway accidents, natural disasters, conflicts, and economic crisis. To do that, we exploit the category hierarchy in Wikipedia where a category involves a group of finer topic subcategories. Given the initial list of five categories, we crawl articles associated with the categories and their descendants (i.e., subcategories, subsubcategories) up to a hierarchy depth of 6. Here, by exploiting the interlinks across languages, we are able to retrieve Wikipedia articles in non-English languages for the chosen categories. In the next step, the crawled articles are then cleaned by removing markup elements (e.g., lists, tables, images). Finally, the articles are split into sentences and tokenized into words by Trankit (M. V. Nguyen, Lai, Pouran Ben Veyseh, & Nguyen, 2021), a multilingual NLP toolkit.

Language	#Docs	#Events	#Rels	#Cross
English	438	8,732	841	8.7%
Danish	519	6,909	904	36.1%
Spanish	746	11,839	545	22.0%
Turkish	1,357	14,179	1,068	64.4%
Urdu	531	4,975	586	27.3%
Total	3,591	46,634	3,944	34.7%

Table 23. Statistics of our mSubEvent dataset. #Rels represents the number of subevent relations while #Cross indicate the percentage of subevent relations that involve event mentions in different sentences.

Annotating Wikipedia articles can be challenging and overwhelming as the articles tend to be long and the number of possible mention pairs grows quadratically with respect to the number of event mentions in a document. As such, to facilitate the annotators, we follow prior practices for event annotation (Ebner et al., 2020; Mostafazadeh et al., 2016) to split the cleaned articles into shorter chunks that contain five consecutive sentences (called documents in this work). In this way, the annotators only need to process a shorter document at a time to improve their attention and quality of annotated data.

Human Annotation: We hire annotators from `upwork.com`, a global crowdsourcing platform. We only consider candidates who are native speakers in our target languages and fluent in English. These information are provided in the annotators' profile in the platform. The candidates are provided with annotation guidelines and instructions for annotation interface, i.e., based on the BRAT annotation tool in our case (Stenetorp et al., 2012). Afterward, the candidates are invited to perform a designed test for both event mention and subevent relation annotation. For each language, the top two candidates are chosen for the annotation job.

We divide our annotation task into two steps for event mention and subevent relation annotation. For each language, we annotate subevent relations over the outputs from event mention annotation (i.e., after event mention annotation has been completed and finalized for all documents). Given a sample of selected documents for a language, for each step, the two annotators for that language independently annotate event mentions/subevent relations for the documents. Each annotator will completely annotation one document at a time. Afterward, the annotation conflicts are presented to the annotators for further discussion and revision to produce the final version of annotated documents for the current task. This helps to achieve high agreement and consistency for our dataset.

Data Analysis: Table 22 shows our Kappa scores for annotation agreements of event mention and subevent relation annotation over five languages. Note that these scores are computed by comparing the independent annotations of the annotators over the documents (i.e., before the discussion to resolve conflicts). As can be seen, the scores are very close to an either substantial or almost perfect agreement for all the tasks and languages, thus demonstrating the high quality of our multilingual SRE dataset. We also find that non-English languages tend to have lower annotation agreement scores for both annotation tasks, thus highlighting the challenges of SRE for non-English languages that necessitate further research effort in this area. In addition, Table 23 shows major statistics. The #Cross column in the table shows that all languages in our dataset involve event mentions in different sentences for the subevent relations (i.e., cross-sentence relation), thus necessitating document-level context modeling. Among the five languages, English has the smallest percentage for cross-sentence relations which further reveals the challenge of SRE for non-English languages.

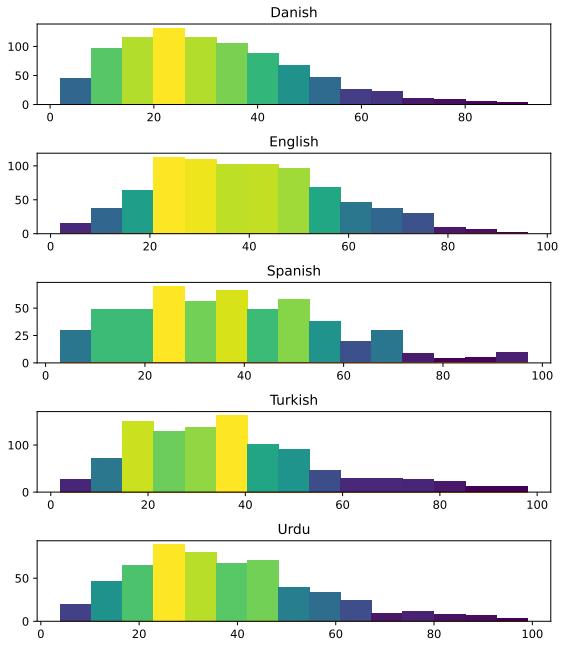

Figure 7. Distributions of distances between two event mentions with subevent relations. Distances are measured via the number of words.

To provide more insight for our multilingual SRE dataset mSubEvent, Figure 7 shows the distributions of distances between two event mentions with subevent relations for five languages in mSubEvent. As can be seen, a majority of event mention pairs are 10 to 50 words away from each other in the documents, suggesting diverse levels of context information between event mentions that must be captured by SRE models for mSubEvent.

6.3 Model

Following prior work (Trong et al., 2022), we utilize pairwise classification to formulate SRE. Given a document $D = [w_1, w_2, \ldots, w_n]$ (of n words) with w_{e_1} and w_{e_2} as two input event mentions/triggers, a SRE model needs to classify the relation between w_{e_1} and w_{e_2} according to one of the three types for subevents, i.e., `PARENT-CHILD`, `CHILD-PARENT`, and `NOREL`. Here, the `NOREL` type is to indicate no subevent relation.

6.3.1 Input Encoding.

In the first step, our model feeds the input document D into a pre-trained language model (PLM), i.e., RoBERTa (Y. Liu et al., 2019), to obtain a representation vector v_i for each word $w_i \in D$. Here, we utilize the hidden vectors in the last transformer layer where vectors for the word-pieces in w_i are averaged to compute v_i. For convenience, let $V = \{v_1, v_2, \ldots, v_n\}$ be the sequence of representation vectors for the words in D. Note that if the length of the input document exceeds the length limit in PLMs (i.e., 512 sub-tokens), we split the document into smaller segments to fit into the limit and run PLM over each segment separately to obtain the representations in V.

6.3.2 Structure Induction.

As presented in the introduction, our method aims to transform the sequential format of D into a graph representation that can better capture important context and structures for representation learning for SRE. Motivated by the dependency path between w_{e_1} and w_{e_2} to capture important context for relation prediction (Gupta et al., 2019; Y. Zhang et al., 2018), we first build a dependency graph T for D to initialize our graph construction process. In particular, we obtain dependency trees for the sentences in the document and connect the roots of the trees for consecutive sentences to create T. We leverage the Trankit toolkit (M. V. Nguyen, Lai, Pouran Ben Veyseh, & Nguyen, 2021) to generate dependency trees and ignore directions in the edges of the trees in the computation. As such, a property of the non-DP words in T is that they can involve both important and irrelevant context words for our subevent prediction problem (as demonstrated in the introduction). Accordingly, to compute an effective graph structure for D for SRE, our goal is to prune the dependency graph T so that only important

context words are retained (i.e., removing irrelevant works). Using in-DP words in T as the anchor (i.e., presumably with important context), we aim to further select non-DP words that involve important context to perform the pruning of T for SRE. To this end, we propose to cast the non-DP word selection problem into an alignment problem between non-DP and in-DP words in which a non-DP word is considered as important for subevent prediction if it is aligned with one in-DP word in the alignment (i.e., extending the anchor in-DP words). To compute the alignment between the words for SRE, we propose to model both syntactic and semantic similarities between non-DP and in-DP words where Optimal Transport (OT) (Peyre & Cuturi, 2019) is leveraged to facilitate the information combination for optimal alignment computation.

6.3.3 Optimal Transport.

Optimal Transport is an established method to find the optimal plan to transform one distribution to another. Given two distributions $p(x)$ and $q(y)$ over discrete domains $\mathcal{X}$ and $\mathcal{Y}$ (respectively), and the cost function $C(x, y) : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}_+$ to map $\mathcal{X}$ into $\mathcal{Y}$, OT finds the optimal joint alignment/distribution $\pi^*(x, y)$ (over $\mathcal{X} \times \mathcal{Y}$) with marginals $p(x)$ and $q(y)$, i.e., the cheapest transportation from $p(x)$ to $q(y)$, by solving the following problem:

$$\begin{gathered} \pi^*(x, y) = \min_{\pi \in \Pi(x,y)} \sum_{\mathcal{Y}} \sum_{\mathcal{X}} \pi(x, y) C(x, y) dx dy \\ \textbf{s.t. } x \sim p(x) \text{ and } y \sim q(y), \end{gathered} \tag{6.1}$$

where $\Pi(x, y)$ involves all joint distributions with marginals $p(x)$ and $q(y)$. Here, the distribution $\pi^*(x, y)$ is a matrix whose entry (x, y) captures the probability of transforming the data point $x \in \mathcal{X}$ to $y \in \mathcal{Y}$ for the conversion of $p(x)$ to $q(y)$. Note that to obtain a hard alignment between data points $\mathcal{X}$ and $\mathcal{Y}$, we can align each

row of $\pi^*(x, y)$ with the column with the highest probability:

$$y^* = \text{argmax}_{y \in \mathcal{Y}} \pi^*(x, y) \forall x \in \mathcal{X}$$

To adopt OT to solve our non-DP word selection problem, we propose to treat the in-DP words in T as the data points for domain $\mathcal{Y}$ while the non-DP words will be used for domain $\mathcal{X}$. As such, OT facilitates the integration of syntactic and semantic similarities into the computation of optimal alignment between in-DP and non-DP words by leveraging these information to compute the transformation cost function $C(x, y)$ and the probability distributions $p(x)$ and $p(y)$. In particular, to compute $p(x)$ and $q(y)$ for $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, we use syntactic distances of the words to the input event mentions. Formally, for each word $w_i \in D$, we obtain the lengths of the paths that connect w_i with the input event mentions w_{e_1} and w_{e_2} in the dependency graph T, i.e., d_i^1 and d_i^2, respectively. The syntactic importance of w_i for SRE is then determined by:

$$syn(w_i) = \max(d_i^1, d_i^2) \tag{6.2}$$

Afterward, the distributions $p(x)$ and $p(y)$ can be obtained by normalizing the syntactic importance scores (with softmax) for the words in the corresponding sets of $\mathcal{X}$ and $\mathcal{Y}$. Next, for the transportation cost $C(x, y)$, we leverage the contextual semantics for the words x and y, measured by the Euclidean distance between their representation vectors v_x and v_y (i.e., in V):

$$C(x, y) = ||v_x - v_y|| \tag{6.3}$$

In addition, to aid the selection of non-DP important words, we introduce an extra data point, called `NIL`, to the in-DP set $\mathcal{Y}$ so non-DP words in $\mathcal{X}$ aligned with `NIL` will be considered irrelevant and excluded from T for graph structure induction for SRE. As such, the representation for `NIL` is computed using average

of the representation vectors of the in-DP words in $\mathcal{Y}$ (i.e., to used for the transportation cost $C(x, y)$). Also, we utilize the average syntactic importance scores for the words in $\mathcal{X}$ to serve as the syntactic score $syn(\texttt{NIL})$ for `NIL` (the distribution $p(x)$ can be obtained accordingly). In this way, solving Equation 6.1 returns the optimal alignment $\pi^*(x, y)$ that can provide hard alignment for the data points in $\mathcal{X}$ and $\mathcal{Y}$[1]. Let I be the subset of non-DP words in $\mathcal{X}$ that are not aligned with `NIL` in $\mathcal{Y}$ according to $\pi^*(x, y)$ (i.e., irrelevant words). To this end, to prune the dependency graph T for SRE, we can eliminate the words in I from T to produce a new graph that only involves induced important context words for subevent prediction. However, as the resulting graph might be disconnected, we further retain the words in the paths between any word in I and the input event mentions (i.e., w_{e_1} and w_{e_2}), generating a new graph T' to serve as our induced graph structure to represent the input document for SRE.

In the next step, given the induced structure T', we feed it into a Graph Convolutional Network (GCN) (Kipf & Welling, 2017; T. H. Nguyen & Grishman, 2018) to learn richer representation vectors for the words in T'. The representation vectors from the PLM (i.e., in V) serve as the inputs for GCN. As such, the induced hidden vectors in the last layer of GCN are denoted by

$$V' = \{v'_{i_1}, \ldots, v'_{i_{|T'|}}\}$$

Finally, we obtain an overall representation vector A for D for SRE via the concatenation:

$$A = [v'_{e_1}, v'_{e_2}, \text{max\_pool}(v'_{i_1}, \ldots, v'_{i_{|T'|}})]$$

[1]We employ the entropy-based approximation of OT and solve it with the Sinkhorn algorithm (Peyre & Cuturi, 2019).

where v'_{e_1} and v'_{e_2} are the GCN-induced representation vectors in V' for the input event mentions w_{e_1} and w_{e_2}. The representation A will then be sent into a feed-forward network FF with softmax in the end to compute a distribution $P(\cdot|D, w_{e_1}, w_{e_2}) = FF(A)$ over the possible subevent relations. The negative log-likelihood function over $P(\cdot|D, w_{e_1}, w_{e_2})$ will be used to train our SRE model in this work.

6.4 Experiments

Datasets: Similar to prior work (Trong et al., 2022; H. Wang et al., 2020; H. Wang, Zhang, Chen, & Roth, 2021), we evaluate our proposed model with optimal transport (called OT-SRE) on the popular datasets for SRE, i.e., **HiEve** (Glavaš et al., 2014) and **Intelligence Community (IC)** (Hovy et al., 2013). In particular, HiEve provides subevent and coreference relation annotation for events over 100 news articles using four relation labels, i.e., `PARENT-CHILD`, `CHILD-PARENT` (for subevents), `COREF` (for coreference), and `NOREL` (for no relation). To make it comparable, we utilize the same data split and setting as the current work with best-reported performance for HiEve (Trong et al., 2022; H. Wang et al., 2020), featuring 80 documents for training (2,423 subevent relations and 0.4 probability for down-sampling of negative examples) and 20 documents for testing (817 subevent relations). For IC, it also annotates 100 news articles for four subevent and coreference relations as in HiEve. Following the same setting in the current state-of-the-art method for IC (H. Wang et al., 2021), we discard relations with implicit event mentions and compute transitive closure for both subevent relations and coreference to obtain annotation for all event mention pairs as in HiEve (Glavaš et al., 2014). Also, IC is divided into three portions with 60/20/20 documents for training/development/test data respectively.

In addition, we evaluate the SRE models on the new multilingual dataset mSubEvent to provide baselines for future research. Here, we randomly split the documents for each language in mSubEvent into three separate parts with a ratio of 3/1/1 for training, development, and test data (respectively). We will use mSubEvent to evaluate SRE models in both monolingual and cross-lingual transfer learning experiments.

Hyper-parameters: We fine-tune the hyper-parameters for our OT-SRE model over English development data of mSubEvent and apply the selected values for all experiments for consistency. In particular, the selected hyper-parameters for our model include: 2 layers for the GCN and feed-forward (i.e., FF) models with 512 dimensions for the hidden vectors, $5e$-5 for the learning rate with Adam optimizer, and 16 for the batch size. Finally, we utilize the the RoBERTa$_{base}$ model (Y. Liu et al., 2019) to encode input texts for HiEve as in prior work (Trong et al., 2022; H. Wang et al., 2020). For mSubEvent, we use the multilingual pre-trained language models (base versions), i.e., mBERT (Devlin et al., 2019) and XLMR (Conneau et al., 2020), for multilingual text encoding.

Baselines: For HiEve, we compare our proposed SRE model with the following baselines using the same data setting: **StructLR** (Glavaš et al., 2014) with feature engineering, **TacoLM** (Zhou, Ning, Khashabi, & Roth, 2020) with temporal common sense knowledge, **Joint** (H. Wang et al., 2020) with joint subevent and temporal relation extraction, **EventSeg** (H. Wang et al., 2021) with event-based text segmentation, and **SCS** (Trong et al., 2022) with the selection of best context sentences for SRE. Similarly, for IC, we consider **Joint**, **EventSeg**, and **SCS** for the baselines. Note that **SCS** and **EventSeg** have the state-of-the-art (SOTA) performance for HiEve and IC (respectively) in the literature. We run the

code for **SCS** (Trong et al., 2022) and **EventSeg** (H. Wang et al., 2021) from the original papers to obtain their performance for IC and HiEve (respectively) for completeness.

6.4.1 Performance Comparison.

Table 24 presents the performance of the models on the test data of HiEve and IC. To be comparable with previous work (Glavaš et al., 2014; Trong et al., 2022), our model is trained for all the four relation labels in HiEve (i.e., including `COREF`); however, the performance for comparison is only measured according to the F1 scores of the subevent relations, i.e., `PARENT-CHILD`, `CHILD-PARENT`, and their micro-average. The most important observation from the table is that the proposed model OT-SRE significantly outperforms all the baseline models ($p < 0.01$) with substantial gaps for both HiEve and IC. In particular, for HiEve, OT-SRE is better than the prior SOTA method SCS by 3% over the average F1 score for subevent relations. OT-SRE is better than the prior SOTA methods for HiEve (i.e., SCS) and IC (i.e., EventSeg) by 3% and 2.7% (respectively) over the average F1 score for subevent relations. These results thus clearly demonstrate the effectiveness of our OT-based approach for graph structure induction to optimize representation learning for SRE.

6.4.2 Multilingual Evaluation.

We further evaluate SRE models over multiple languages using the mSubEvent dataset. We employed the best baselines, i.e., EventSeg and SCS, from Table 24 in this experiment. In addition, for reference, we report the performance of the **PLM** model that directly uses the representation vectors learned by the multilingual PLMs (i.e., in V) to form the overall representations for subevent prediction, i.e., $A = [v_{e_1}, v_{e_2}, \text{max\_pool}(v_1, \ldots, v_n)]$. As such, we first explore

Model	F-score		
	PC	**CP**	**Avg**
HiEve			
StructLR (Glavaš et al., 2014)	52.2	63.4	57.7
TacoLM (Zhou et al., 2020)	48.5	49.4	48.9
Joint (H. Wang et al., 2020)	62.5	56.4	59.5
EventSeg (H. Wang et al., 2021)	58.6	57.9	58.3
SCS (Trong et al., 2022)	68.7	63.2	65.9
OT-SRE (ours)	**70.3**	**67.4**	**68.9**
IC			
(Araki et al., 2014)	-	-	26.2
Joint (H. Wang et al., 2020)	42.1	49.5	45.8
EventSeg (H. Wang et al., 2021)	44.6	51.6	48.1
SCS (Trong et al., 2022)	47.5	51.8	49.7
OT-SRE (ours)	**48.9**	**52.6**	**50.8**

Table 24. Model performance on test data of HiEve and IC datasets. We focus on the performance for `PARENT-CHILD` (PC), `CHILD-PARENT` (CP), and their micro-average to be consistent with prior evaluation for SRE.

monolingual learning settings where models are trained and tested on data of the same language. In particular, Table 25 shows the monolingual performance of the SRE models for five languages in mSubEvent when either mBERT or XLMR is used for multilingual text encoding. As can be seen, OT-SRE is also significantly better than all baseline models over different languages in mSubEvent, thus highlighting the ability to generalize to different languages of the OT-induced graph structures for SRE. Importantly, we find that the performance of the models over mSubEvent is still far from being satisfactory (i.e., much worse than that for HiEve). Future research will have ample opportunities to improve the performance on mSubEvent.

In addition, Table 26 investigates model performance in the cross-lingual transfer learning setting where models are trained over English training data (i.e., the source language) and directly evaluated on test data of other languages (i.e.,

Model	English	Danish	Spanish	Turkish	Urdu
mBERT					
PLM	36.5	30.2	23.6	39.0	34.1
EventSeg	41.1	41.7	37.4	42.8	43.1
SCS	46.8	45.9	40.6	44.0	50.1
OT-SRE	**49.3**	**48.9**	**42.1**	**50.1**	**52.2**
XLMR					
PLM	40.1	33.1	34.9	41.9	45.2
EventSeg	42.3	40.0	41.3	42.9	51.1
SCS	48.1	41.8	**43.2**	45.1	51.6
OT-SRE	**49.5**	**50.0**	42.7	**52.2**	**52.4**

Table 25. Model performance (F-scores) for monolingual settings in mSubEvent.

the target languages). It is clear from the table that the cross-lingual performance in Table 26 is inferior to the English monolingual performance in Table 25, thus emphasizing the challenge of cross-lingual knowledge transfer for subevent recognition for future work. Finally, Table 26 further demonstrates better ability to learn transferable representations across languages of OT-SRE to yield the best cross-lingual performance for SRE. We attribute this to the advantages of the induced graph structures to represent input texts in OT-SRE that can be more general across languages than the sequential text order in the baseline methods.

Model	Danish	Spanish	Turkish	Urdu
mBERT				
PLM	23.6	22.6	13.5	11.7
EventSeg	29.0	32.2	16.5	16.4
SCS	**34.6**	36.4	18.9	19.9
OT-SRE	33.1	**37.1**	**19.0**	**27.4**
XLMR				
PLM	25.1	25.4	17.4	18.4
EventSeg	28.5	31.3	20.9	21.4
SCS	41.2	33.7	19.3	22.5
OT-SRE	**42.8**	**34.4**	**22.6**	**26.0**

Table 26. Cross-lingual performance (F-score) on mSubEvent with English as the source language. The language in each column indicates the target languages.

6.4.3 Ablation Study.

We study the ablated models of OT-SRE to understand the contribution of the designed components in the our model. Table 27 reports the performance over test data of HiEve for the ablation study. In particular, lines 2 and 3 in the table indicate the baselines where the OT component is not included to induce the graph structure T' for input document. Instead, the DP between the event mentions (i.e., in line 2 with **-OT**) or the full dependency graph T (i.e., in line 3 with **- Pruning**) is leveraged as the graph structure for representation learning. As can be seen, both lines 2 and 3 lead to significantly worse performance for ST-SRE, thus demonstrating the importance of the OT component to induce optimal graph structures to represent input texts for SRE.

ID	Model	**CP**	**PC**	**Avg.**
1	**OT-SRE (full)**	**70.3**	**67.4**	**68.9**
2	- OT	67.8	62.2	65.0
3	- Pruning	60.3	65.8	63.1
4	- GCN	64.3	67.6	66.0
5	- OT-GCN	63.7	57.1	60.4
6	- Syntax in OT	69.1	65.7	67.4
7	- Semantic in OT	65.3	66.8	66.1
8	- DP	69.1	67.2	68.2

Table 27. Ablation study on HiEve test data. We report the the performance for `PARENT-CHILD` (PC), `CHILD-PARENT` (CP), and their micro-average.

In addition, in lines 4 and 5, we study variants of OT-SRE that eliminates the GCN component. In particular, in line 4 with **- GCN**, we still employ the OT component to compute the graph structure T'; however, instead of using GCN-induced representations, the overall representation for prediction is computed over PLM-induced representations in V, i.e., $A = [v_{e_1}, v_{e_2}, \text{max\_pool}(v_j | w_j \in T')]$ where the max-pooling is done for the words in the computed graph structure T'. For

line 5 with **- OT-GCN**, both the OT and GCN components are removed from OT-SRE. The overall representation is thus also computed with the PLM-induced representations V, i.e., $A = [v_{e_1}, v_{e_2}, \text{max\_pool}(v_j | w_j \in D)]$, using a max-pooling operation over the entire input text D. It is clear from the table that GCN is helpful to learn better representations for SRE as removing it will significantly hurts the performance for OT-SRE in both lines 4 and 5.

Further, line 6 (**- Syntax in OT**) evaluates OT-SRE when syntactic information (i.e., the important scores $syn(w_i)$) is not used to obtain the domain distributions $p(x)$ and $p(y)$ in the OT component. Instead, uniform distributions are leveraged for $p(x)$ and $p(y)$ in this case. Also, for line 7 (**- Semantic in OT**), this variant avoids semantic information with contextual representations in V to compute the transformation cost $C(x, y)$ for OT. Instead, it employs a simple constant cost function $C(x, y) = 1$. As such, the superior performance of OT-SRE over these ablated models shows that both syntactic and semantic information are critical for the OT component to ensure the best performance for OT-SRE. Finally, in line 8 (i.e., **- DP**), our OT-SRE model only includes the two input event mentions/triggers in domain (Y). As such, domain $\mathcal{X}$ for alignment in OT will contain all other words in D, including the words on the dependency path. The worse performance in line 8 shows that only using event mentions as the anchor for OT alignment is not optimal, necessitating dependency paths to provide better starting points to extend to effective graph structures for SRE.

6.4.4 Case Study.

We perform a case study to analyze the examples in HiEve that can be successfully predicted by OT-SRE, but fail the baseline without OT (i.e., in line 2 of Table 27 to directly use DP for representation). A major observation in our

analysis is that OT-SRE can find important context words beyond the DP to aid subevent prediction.

For example, consider the following sentence:

> *"Over 90 Palestinians and one Israeli soldier have been* ***killed*** ***since*** *Israel launched a massive* ***offensive*** *into the Gaza Strip on June 28."*

with "*killed*" and "*offensive*" as the event mentions. While the DP "*killed* $\rightarrow$ *launched* $\rightarrow$ *offensive* does not provide clear context information to recognize the subevent relation, our OT-SRE is able to align the DP with the word "*since*" to facilitate SRE.

A similar example can be found in the following sentence:

> "*No one has been arrested over Sunday's* ***attack*** *in Kabul and the Taliban have denied any involvement. Arsala Rahmani has been* ***killed*** *by enemies of Afghanistan. Both NATO and the US embassy in Kabul have also condemned the* ***assassination****."*

with the event mentions "*attack*" and "*killed*". The important context word "*assassination*" does not belong to the DP between the event mentions, but it is successfully included in the graph structure by OT-SRE for correct prediction.

6.5 Related Work

Early methods for SRE have exploited various contextual features for input texts (i.e., feature engineering) for machine learning models (Aldawsari & Finlayson, 2019; Araki et al., 2014; Glavaš et al., 2014). To alleviate feature engineering, recent works have explored deep learning models to induce representations for SRE from data, introducing joint inference with temporal relations (H. Wang et al., 2020; Zhou et al., 2020) and large PLMs (Trong et

al., 2022; H. Wang et al., 2021; Yao, Dai, Ramaswamy, Min, & Huang, 2020). Existing datasets for SRE include HiEve (Glavaš et al., 2014), IC (Araki et al., 2014; Hovy et al., 2013), and RED (O'Gorman et al., 2016). However, none of such methods and datasets considers graph structure induction for input texts and multilingual learning for SRE as we do. Regarding related work on event-event relation extraction, we also note recent studies for other types of relations between events, including causal (Caselli & Vossen, 2017; Man, Nguyen, & Nguyen, 2022; Tran Phu & Nguyen, 2021; Zuo et al., 2020), coreference (Choubey, Lee, Huang, & Wang, 2020; Minh Tran, Phung, & Nguyen, 2021; T. H. Nguyen et al., 2016g; Phung et al., 2021), and temporal (Ning, Feng, & Roth, 2017; Tran Phu, Nguyen, & Nguyen, 2021) relations. Finally, optimal transport has also been recently used to solve NLP problems (Guzman-Nateras, Nguyen, & Nguyen, 2022; Pouran Ben Veyseh & Nguyen, 2022); however, none of the previous work has employed OT for subevent relation extraction as we do.

6.6 Summary

- We present a novel method for subevent relation extraction that leverages optimal transport to induce effective graph structures for input texts to improve representation learning. The graph structure representation is able to directly capture important context words and their connections to facilitate SRE.

- We introduce the first multilingual dataset for SRE that provides human annotation for five languages with high quality. Extensive experiments demonstrate the effectiveness of our method with state-of-the-art performance on different datasets and learning settings. Our new dataset also offers ample

opportunities for future research. In the future, we plan to extend our method and dataset to other event-event relations.

CHAPTER VII

CONCLUSION

7.1 Summary

The main target of this dissertation is to advance the field of Low-Resource Event Extraction through a holistic set of methods including designing neural network architecture to integrate external resources, developing efficient training signals under limited supervision, and creating new resources for future research.

First, we designed language-agnostic model architectures to enhance the representation learning of the event detection task. We proposed a gating mechanism to filter out information for the trigger candidate for the existing event detection models based on graph convolutional neural networks. Furthermore, to incorporate external resources such as syntactic features derived from the dependency graph of the sentence, we designed novel network architectures and auxiliary loss functions to enrich the information and reduce noisy information induced in the representation for event detection.

Second, we developed novel training methods to efficiently use limited supervision in few-shot learning for event detection. Under limited training supervision for new classes, we transfer the knowledge from the existing knowledge bases such as word sense disambiguation corpus to provide the model with more supervision from related tasks, hence, helping the few-shot learning model to generalize better on unseen data. Moreover, we tackled the poor sampling problem during the training time of few-shot learning for event detection by encouraging interaction between data samples, resulting in richer prototypes for the prototypical network. Our prediction consistency across seen samples also make the model more robust to noise during the training of the few-shot learning model. This results in

a significant improvement of the few-shot learning model without any additional supervision in the inference time.

Third, due to the scarcity of benchmark corpora for non-English languages, we created the first multilingual corpus for event-event relation extraction with a focus on causality and sub-event relations. These corpora created research opportunities for event extraction and event relation extraction for low-resource languages. Subsequently, we hope to expand the coverage of language technologies to the broader non-English-spoken population, hence, democratizing access to language technologies to more people in the world.

Finally, we showed that language-agnostic features help transfer knowledge across languages for event-event relation extraction. In particular, our experiment shows that structural features derived from dependency graphs are easily transferable across languages. Moreover, language-agnostic context selection methods like optimal transport can alleviate the effect of noisy information appearing in all examined languages.

7.2 Limitation

Throughout this dissertation, the methods were built with dependency on other toolkits and models such as dependency parser M. V. Nguyen, Lai, Pouran Ben Veyseh, and Nguyen (2021) and large pre-trained language model Conneau et al. (2020). Hence, these methods only apply to languages that have a dependency parser and a large pre-trained language model. Unfortunately, only a few hundred popular languages have both a dependency parser and a pre-trained LLM. In other words, even though these methods can be used for many languages, it is not a universal method for every language, especially extremely low-resource languages.

7.3 Future work

This dissertation has provided a broad spectrum of topics and methods to solve low-resource event extraction, however, there are many other potential research topics and methods that have yet to be explored.

Even though the event extraction task has been studied for more than two decades and the accuracy of event extraction is getting improved every year, the application of event extraction in real life is still subpar compared to what has been observed in other tasks such as machine translation and sentiment analysis. There is still a large gap between how the event task is currently formulated and what people want to achieve in their real-life tasks. We believe this gap can be bridged with more research focus on higher-level tasks such as event timelining Minard et al. (2015), event summarization Steen and Markert (2019), and more complex functionality on top of events such as reasoning on knowledge graph X. Wu, Huang, Fung, and Ji (2022).

The advancement of large language models has brought in new potential capabilities for event extraction that allows expanding event extraction potential to new horizons. Firstly, these models now possess the ability to process an almost limitless amount of context, thanks to optimization that has significantly reduced their compute requirements Press, Smith, and Lewis (2021). This breakthrough enables them to handle extensive information seamlessly. As such, event extraction can significantly benefit from it, as the model now has access to all the available context, expectedly, producing much more precise answers. Secondly, large language models have undergone specialized training to swiftly comprehend tasks based on their descriptions Ouyang et al. (2022). Consequently, the need for explicit task formulations, such as sequence labeling with BIO tags, has diminished.

This development paves the way for incorporating event extraction expertise into various applications, including question-answering, virtual assistants, and countless other tools utilized in our daily lives. Third, the large language models are usually trained on a large multilingual text corpus Xue et al. (2021), inherently forcing the large language model's multilingual capability Brown et al. (2020) such as translating, understanding, and answering questions in other languages. This allows the EE models built on top of these new large language models to be able to work with a wide range of languages with minimal modification.

REFERENCES CITED

Ahn, D. (2006, July). The stages of event extraction. In *Proceedings of the workshop on annotating and reasoning about time and events* (pp. 1–8). Sydney, Australia: Association for Computational Linguistics. Retrieved from https://aclanthology.org/W06-0901

Aldawsari, M., & Finlayson, M. (2019, July). Detecting subevents using discourse and narrative features. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 4780–4790). Florence, Italy: Association for Computational Linguistics. Retrieved from https://aclanthology.org/P19-1471 doi: 10.18653/v1/P19-1471

Araki, J., Liu, Z., Hovy, E., & Mitamura, T. (2014, May). Detecting subevent structure for event coreference resolution. In *Proceedings of the ninth international conference on language resources and evaluation (LREC'14).* Reykjavik, Iceland: European Language Resources Association (ELRA). Retrieved from http://www.lrec-conf.org/proceedings/lrec2014/pdf/963_Paper.pdf

Araki, J., & Mitamura, T. (2015, September). Joint event trigger identification and event coreference resolution with structured perceptron. In *Proceedings of the 2015 conference on empirical methods in natural language processing* (pp. 2074–2080). Lisbon, Portugal: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D15-1247 doi: 10.18653/v1/D15-1247

Baker, C. F., Fillmore, C. J., & Lowe, J. B. (1998, August). The Berkeley FrameNet project. In *36th annual meeting of the association for computational linguistics and 17th international conference on computational linguistics, volume 1* (pp. 86–90). Montreal, Quebec, Canada: Association for Computational Linguistics. Retrieved from https://aclanthology.org/P98-1013 doi: 10.3115/980845.980860

Bao, Y., Wu, M., Chang, S., & Barzilay, R. (2020). Few-shot text classification with distributional signatures. In *Proceedings of the international conference on learning representations (ICLR).*

Beltagy, I., Lo, K., & Cohan, A. (2019, November). SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)* (pp. 3615–3620). Hong Kong, China: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D19-1371 doi: 10.18653/v1/D19-1371

Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. In *Journal of machine learning research.*

Berant, J., Srikumar, V., Chen, P.-C., Vander Linden, A., Harding, B., Huang, B., ... Manning, C. D. (2014, October). Modeling biological processes for reading comprehension. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1499–1510). Doha, Qatar: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D14-1159 doi: 10.3115/v1/D14-1159

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., & Taylor, J. (2008). Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 acm sigmod international conference on management of data* (pp. 1247–1250).

Bronstein, O., Dagan, I., Li, Q., Ji, H., & Frank, A. (2015, July). Seed-based event trigger labeling: How far can event descriptions get us? In *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)* (pp. 372–376). Beijing, China: Association for Computational Linguistics. Retrieved from https://aclanthology.org/P15-2061 doi: 10.3115/v1/P15-2061

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

Caselli, T., & Vossen, P. (2017, August). The event StoryLine corpus: A new benchmark for causal and temporal relation extraction. In *Proceedings of the events and stories in the news workshop* (pp. 77–86). Vancouver, Canada: Association for Computational Linguistics. Retrieved from https://aclanthology.org/W17-2711 doi: 10.18653/v1/W17-2711

Cañete, J., Chaperon, G., Fuentes, R., Ho, J.-H., Kang, H., & Pérez, J. (2020). Spanish pre-trained bert model and evaluation data. In *Pml4dc at iclr 2020.*

Chen, J., Lin, H., Han, X., & Sun, L. (2021, November). Honey or poison? solving the trigger curse in few-shot event detection via causal intervention. In *Proceedings of the 2021 conference on empirical methods in natural language processing* (pp. 8078–8088). Online and Punta Cana, Dominican Republic: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.emnlp-main.637 doi: 10.18653/v1/2021.emnlp-main.637

Chen, Y., Liu, S., He, S., Liu, K., & Zhao, J. (2016). Event extraction via bidirectional long short-term memory tensor neural networks. In *Chinese computational linguistics and natural language processing based on naturally annotated big data* (pp. 190–203). Springer.

Chen, Y., Liu, S., Zhang, X., Liu, K., & Zhao, J. (2017, July). Automatically labeled data generation for large scale event extraction. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 409–419). Vancouver, Canada: Association for Computational Linguistics. Retrieved from https://aclanthology.org/P17-1038 doi: 10.18653/v1/P17-1038

Chen, Y., Xu, L., Liu, K., Zeng, D., & Zhao, J. (2015, July). Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)* (pp. 167–176). Beijing, China: Association for Computational Linguistics. Retrieved from https://aclanthology.org/P15-1017 doi: 10.3115/v1/P15-1017

Chen, Y., Yang, H., Liu, K., Zhao, J., & Jia, Y. (2018, October-November). Collective event detection via a hierarchical and bias tagging networks with gated multi-level attention mechanisms. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 1267–1276). Brussels, Belgium: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D18-1158 doi: 10.18653/v1/D18-1158

Chen, Z., & Ji, H. (2009, June). Can one language bootstrap the other: A case study on event extraction. In *Proceedings of the NAACL HLT 2009 workshop on semi-supervised learning for natural language processing* (pp. 66–74). Boulder, Colorado: Association for Computational Linguistics. Retrieved from https://aclanthology.org/W09-2209

Chi, Z., Dong, L., Ma, S., Huang, S., Singhal, S., Mao, X.-L., ... Wei, F. (2021, November). mT6: Multilingual pretrained text-to-text transformer with translation pairs. In *Proceedings of the 2021 conference on empirical methods in natural language processing* (pp. 1671–1683). Online and Punta Cana, Dominican Republic: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.emnlp-main.125 doi: 10.18653/v1/2021.emnlp-main.125

Cho, K., van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014, October). On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, eighth workshop on syntax, semantics and structure in statistical translation* (pp. 103–111). Doha, Qatar: Association for Computational Linguistics. Retrieved from https://aclanthology.org/W14-4012 doi: 10.3115/v1/W14-4012

Choubey, P. K., Lee, A., Huang, R., & Wang, L. (2020, July). Discourse as a function of event: Profiling discourse structure in news articles around the main event. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 5374–5386). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2020.acl-main.478 doi: 10.18653/v1/2020.acl-main.478

Cong, X., Cui, S., Yu, B., Liu, T., Yubin, W., & Wang, B. (2021, August). Few-Shot Event Detection with Prototypical Amortized Conditional Random Field. In *Findings of the association for computational linguistics: Acl-ijcnlp 2021* (pp. 28–40). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.findings-acl.3 doi: 10.18653/v1/2021.findings-acl.3

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., ... Stoyanov, V. (2020, July). Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 8440–8451). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2020.acl-main.747 doi: 10.18653/v1/2020.acl-main.747

Cui, S., Yu, B., Liu, T., Zhang, Z., Wang, X., & Shi, J. (2020, November). Edge-enhanced graph convolution networks for event detection with syntactic relation. In *Findings of the association for computational linguistics: Emnlp 2020* (pp. 2329–2339). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2020.findings-emnlp.211 doi: 10.18653/v1/2020.findings-emnlp.211

Cybulska, A., & Vossen, P. (2014). *Guidelines for ecb+ annotation of events and their coreference.* Technical Report NWR-2014-1, VU University Amsterdam. Retrieved from http://www.newsreader-project.eu/files/2013/01/NWR-2014-1.pdf

Deng, S., Zhang, N., Kang, J., Zhang, Y., Zhang, W., & Chen, H. (2020). Meta-learning with dynamic-memory-based prototypical network for few-shot event detection. In *Proceedings of the 13th international conference on web search and data mining* (pp. 151–159).

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019, June). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)* (pp. 4171–4186). Minneapolis, Minnesota: Association for Computational Linguistics. Retrieved from https://aclanthology.org/N19-1423 doi: 10.18653/v1/N19-1423

Ding, X., Song, F., Qin, B., & LIU, T. (2011). Research on typical event extraction method in the field of music. *Journal of Chinese Information Processing*, *2*.

Do, Q., Chan, Y. S., & Roth, D. (2011, July). Minimally supervised event causality identification. In *Proceedings of the 2011 conference on empirical methods in natural language processing* (pp. 294–303). Edinburgh, Scotland, UK.: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D11-1027

Du, X., & Cardie, C. (2020, July). Document-level event role filler extraction using multi-granularity contextualized encoding. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 8010–8020). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2020.acl-main.714 doi: 10.18653/v1/2020.acl-main.714

Duan, S., He, R., & Zhao, W. (2017, November). Exploiting document level information to improve event detection via recurrent neural networks. In *Proceedings of the eighth international joint conference on natural language processing (volume 1: Long papers)* (pp. 352–361). Taipei, Taiwan: Asian Federation of Natural Language Processing. Retrieved from https://aclanthology.org/I17-1036

Dunietz, J., Levin, L., & Carbonell, J. (2017, April). The BECauSE corpus 2.0: Annotating causality and overlapping relations. In *Proceedings of the 11th linguistic annotation workshop* (pp. 95–104). Valencia, Spain: Association for Computational Linguistics. Retrieved from https://aclanthology.org/W17-0812 doi: 10.18653/v1/W17-0812

Dutta, S., Ma, L., Saha, T. K., Liu, D., Tetreault, J., & Jaimes, A. (2021, June). GTN-ED: Event detection using graph transformer networks. In *Proceedings of the fifteenth workshop on graph-based methods for natural language processing (textgraphs-15)* (pp. 132–137). Mexico City, Mexico: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.textgraphs-1.13 doi: 10.18653/v1/2021.textgraphs-1.13

Ebner, S., Xia, P., Culkin, R., Rawlins, K., & Van Durme, B. (2020, July). Multi-sentence argument linking. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 8057–8077). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2020.acl-main.718 doi: 10.18653/v1/2020.acl-main.718

Ellis, J., Getman, J., Fore, D., Kuster, N., Song, Z., Bies, A., & Strassel, S. M. (2015). Overview of linguistic resources for the tac kbp 2015 evaluations: Methodologies and results. In *Tac.*

Fei, N., Lu, Z., Xiang, T., & Huang, S. (2020). MELR: Meta-learning via modeling episode-level relationships for few-shot learning. In *International conference on learning representations (ICLR).*

Feng, X., Huang, L., Tang, D., Ji, H., Qin, B., & Liu, T. (2016, August). A language-independent neural network for event detection. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers)* (pp. 66–71). Berlin, Germany: Association for Computational Linguistics. Retrieved from https://aclanthology.org/P16-2011 doi: 10.18653/v1/P16-2011

Ferguson, J., Lockard, C., Weld, D., & Hajishirzi, H. (2018, June). Semi-supervised event extraction with paraphrase clusters. In *Proceedings of the 2018 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 2 (short papers)* (pp. 359–364). New Orleans, Louisiana: Association for Computational Linguistics. Retrieved from https://aclanthology.org/N18-2058 doi: 10.18653/v1/N18-2058

Filatova, E., & Hatzivassiloglou, V. (2004, July). Event-based extractive summarization. In *Text summarization branches out* (pp. 104–111). Barcelona, Spain: Association for Computational Linguistics. Retrieved from https://aclanthology.org/W04-1017

Finn, C., Abbeel, P., & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning (ICML)* (pp. 1126–1135).

Fritzler, A., Logacheva, V., & Kretov, M. (2019). Few-shot classification in named entity recognition task. In *Proceedings of the 34th acm/sigapp symposium on applied computing* (pp. 993–1000).

Gao, L., Choubey, P. K., & Huang, R. (2019, June). Modeling document-level causal structures for event causal relation identification. In *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)* (pp. 1808–1817). Minneapolis, Minnesota: Association for Computational Linguistics. Retrieved from https://aclanthology.org/N19-1179 doi: 10.18653/v1/N19-1179

Gao, T., Han, X., Liu, Z., & Sun, M. (2019). Hybrid attention-based prototypical networks for noisy few-shot relation classification. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 33, pp. 6407–6414).

Ge, T., Cui, L., Chang, B., Sui, Z., Wei, F., & Zhou, M. (2018, May). EventWiki: A knowledge base of major events. In *Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018).* Miyazaki, Japan: European Language Resources Association (ELRA). Retrieved from https://aclanthology.org/L18-1079

Ghaeini, R., Fern, X., Huang, L., & Tadepalli, P. (2016, August). Event nugget detection with forward-backward recurrent neural networks. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers)* (pp. 369–373). Berlin, Germany: Association for Computational Linguistics. Retrieved from https://aclanthology.org/P16-2060 doi: 10.18653/v1/P16-2060

Glavaš, G., & Šnajder, J. (2013, October). Event-centered information retrieval using kernels on event graphs. In *Proceedings of TextGraphs-8 graph-based methods for natural language processing* (pp. 1–5). Seattle, Washington, USA: Association for Computational Linguistics. Retrieved from https://aclanthology.org/W13-5001

Glavaš, G., Šnajder, J., Moens, M.-F., & Kordjamshidi, P. (2014, May). HiEve: A corpus for extracting event hierarchies from news stories. In *Proceedings of the ninth international conference on language resources and evaluation (LREC'14)* (pp. 3678–3683). Reykjavik, Iceland: European Language Resources Association (ELRA). Retrieved from http://www.lrec-conf.org/proceedings/lrec2014/pdf/1023_Paper.pdf

Grishman, R., & Sundheim, B. (1996). Message Understanding Conference- 6: A brief history. In *COLING 1996 volume 1: The 16th international conference on computational linguistics.* Retrieved from https://aclanthology.org/C96-1079

Grishman, R., Westbrook, D., & Meyers, A. (2005). Nyu's english ace 2005 system description. In *Ace 2005 evaluation workshop.*

Guo, J., Che, W., Wang, H., Liu, T., & Xu, J. (2016, December). A unified architecture for semantic role labeling and relation classification. In *Proceedings of COLING 2016, the 26th international conference on computational linguistics: Technical papers* (pp. 1264–1274). Osaka, Japan: The COLING 2016 Organizing Committee. Retrieved from https://aclanthology.org/C16-1120

Gupta, P., Rajaram, S., Schütze, H., & Runkler, T. (2019). Neural relation extraction within and across sentence boundaries. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 33, pp. 6513–6520).

Guzman-Nateras, L., Nguyen, M. V., & Nguyen, T. (2022, July). Cross-lingual event detection via optimized adversarial training. In *Proceedings of the 2022 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 5588–5599). Seattle, United States: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2022.naacl-main.409 doi: 10.18653/v1/2022.naacl-main.409

Hadiwinoto, C., Ng, H. T., & Gan, W. C. (2019, November). Improved word sense disambiguation using pre-trained contextualized word representations. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)* (pp. 5297–5306). Hong Kong, China: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D19-1533 doi: 10.18653/v1/D19-1533

Han, R., Ning, Q., & Peng, N. (2019, November). Joint event and temporal relation extraction with shared representations and structured prediction. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)* (pp. 434–444). Hong Kong, China: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D19-1041 doi: 10.18653/v1/D19-1041

Han, X., Zhu, H., Yu, P., Wang, Z., Yao, Y., Liu, Z., & Sun, M. (2018, October-November). FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 4803–4809). Brussels, Belgium: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D18-1514 doi: 10.18653/v1/D18-1514

Hashimoto, C. (2019, November). Weakly supervised multilingual causality extraction from Wikipedia. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)* (pp. 2988–2999). Hong Kong, China: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D19-1296 doi: 10.18653/v1/D19-1296

Hashimoto, C., Torisawa, K., Kloetzer, J., Sano, M., Varga, I., Oh, J.-H., & Kidawara, Y. (2014, June). Toward future scenario generation: Extracting event causality exploiting semantic relation, context, and association features. In *Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 987–997). Baltimore, Maryland: Association for Computational Linguistics. Retrieved from https://aclanthology.org/P14-1093 doi: 10.3115/v1/P14-1093

Hidey, C., & McKeown, K. (2016, August). Identifying causal relations using parallel Wikipedia articles. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 1424–1433). Berlin, Germany: Association for Computational Linguistics. Retrieved from https://aclanthology.org/P16-1135 doi: 10.18653/v1/P16-1135

Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. *Proceedings of the NeurIPS Deep Learning and Representation Learning Workshop*.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735-1780. doi: 10.1162/neco.1997.9.8.1735

Hong, Y., Zhang, J., Ma, B., Yao, J., Zhou, G., & Zhu, Q. (2011, June). Using cross-entity inference to improve event extraction. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies* (pp. 1127–1136). Portland, Oregon, USA: Association for Computational Linguistics. Retrieved from https://aclanthology.org/P11-1113

Hovy, E., Mitamura, T., Verdejo, F., Araki, J., & Philpot, A. (2013, June). Events are not simple: Identity, non-identity, and quasi-identity. In *Workshop on events: Definition, detection, coreference, and representation* (pp. 21–28). Atlanta, Georgia: Association for Computational Linguistics. Retrieved from https://aclanthology.org/W13-1203

Hsi, A., Yang, Y., Carbonell, J., & Xu, R. (2016, December). Leveraging multilingual training for limited resource event extraction. In *Proceedings of COLING 2016, the 26th international conference on computational linguistics: Technical papers* (pp. 1201–1210). Osaka, Japan: The COLING 2016 Organizing Committee. Retrieved from https://aclanthology.org/C16-1114

Hsu, I.-H., Huang, K.-H., Boschee, E., Miller, S., Natarajan, P., Chang, K.-W., & Peng, N. (2022, July). DEGREE: A data-efficient generation-based event extraction model. In *Proceedings of the 2022 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 1890–1908). Seattle, United States: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2022.naacl-main.138 doi: 10.18653/v1/2022.naacl-main.138

Hu, Z., & Walker, M. (2017, August). Inferring narrative causality between event pairs in films. In *Proceedings of the 18th annual SIGdial meeting on discourse and dialogue* (pp. 342–351). Saarbrücken, Germany: Association for Computational Linguistics. Retrieved from https://aclanthology.org/W17-5540 doi: 10.18653/v1/W17-5540

Huang, K.-H., & Peng, N. (2021, June). Document-level event extraction with efficient end-to-end learning of cross-event dependencies. In *Proceedings of the third workshop on narrative understanding* (pp. 36–47). Virtual: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.nuse-1.4 doi: 10.18653/v1/2021.nuse-1.4

Huang, K.-H., Yang, M., & Peng, N. (2020, November). Biomedical event extraction with hierarchical knowledge graphs. In *Findings of the association for computational linguistics: Emnlp 2020* (pp. 1277–1285). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2020.findings-emnlp.114 doi: 10.18653/v1/2020.findings-emnlp.114

Huang, L., Cassidy, T., Feng, X., Ji, H., Voss, C. R., Han, J., & Sil, A. (2016, August). Liberal event extraction and event schema induction. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 258–268). Berlin, Germany: Association for Computational Linguistics. Retrieved from https://aclanthology.org/P16-1025 doi: 10.18653/v1/P16-1025

Huang, L., Ji, H., Cho, K., Dagan, I., Riedel, S., & Voss, C. (2018, July). Zero-shot transfer learning for event extraction. In *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 2160–2170). Melbourne, Australia: Association for Computational Linguistics. Retrieved from https://aclanthology.org/P18-1201 doi: 10.18653/v1/P18-1201

Huang, P., Zhao, X., Takanobu, R., Tan, Z., & Xiao, W. (2020, December). Joint event extraction with hierarchical policy network. In *Proceedings of the 28th international conference on computational linguistics* (pp. 2653–2664). Barcelona, Spain (Online): International Committee on Computational Linguistics. Retrieved from https://aclanthology.org/2020.coling-main.239 doi: 10.18653/v1/2020.coling-main.239

Huang, R., & Riloff, E. (2012). Modeling textual cohesion for event extraction. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 26, pp. 1664–1670).

Huang, Y., & Jia, W. (2021, November). Exploring sentence community for document-level event extraction. In *Findings of the association for computational linguistics: Emnlp 2021* (pp. 340–351). Punta Cana, Dominican Republic: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.findings-emnlp.32 doi: 10.18653/v1/2021.findings-emnlp.32

Jagannatha, A. N., & Yu, H. (2016, June). Bidirectional RNN for medical event detection in electronic health records. In *Proceedings of the 2016 conference of the north American chapter of the association for computational linguistics: Human language technologies* (pp. 473–482). San Diego, California: Association for Computational Linguistics. Retrieved from https://aclanthology.org/N16-1056 doi: 10.18653/v1/N16-1056

Jang, E., Gu, S., & Poole, B. (2017). Categorical reparameterization with gumbel-softmax. *ICLR*.

Ji, H., & Grishman, R. (2008, June). Refining event extraction through cross-document inference. In *Proceedings of acl-08: Hlt* (pp. 254–262). Columbus, Ohio: Association for Computational Linguistics. Retrieved from https://aclanthology.org/P08-1030

Ji, H., Nothman, J., Dang, H. T., & Hub, S. I. (2016). Overview of tac-kbp2016 tri-lingual edl and its impact on end-to-end cold-start kbp. *Proceedings of TAC*.

Joulin, A., Bojanowski, P., Mikolov, T., Jégou, H., & Grave, E. (2018, October-November). Loss in translation: Learning bilingual word mapping with a retrieval criterion. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 2979–2984). Brussels, Belgium: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D18-1330 doi: 10.18653/v1/D18-1330

Kadowaki, K., Iida, R., Torisawa, K., Oh, J.-H., & Kloetzer, J. (2019, November). Event causality recognition exploiting multiple annotators' judgments and background knowledge. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)* (pp. 5816–5822). Hong Kong, China: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D19-1590 doi: 10.18653/v1/D19-1590

Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014, June). A convolutional neural network for modelling sentences. In *Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 655–665). Baltimore, Maryland: Association for Computational Linguistics. Retrieved from https://aclanthology.org/P14-1062 doi: 10.3115/v1/P14-1062

Kim, J.-D., Ohta, T., Pyysalo, S., Kano, Y., & Tsujii, J. (2009, June). Overview of BioNLP'09 shared task on event extraction. In *Proceedings of the BioNLP 2009 workshop companion volume for shared task* (pp. 1–9). Boulder, Colorado: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/W09-1401

Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *ICLR.*

Kodelja, D., Besançon, R., & Ferret, O. (2019). Exploiting a more global context for event detection through bootstrapping. In *European conference on information retrieval* (pp. 763–770).

Lai, V., Dernoncourt, F., & Nguyen, T. H. (2021, November). Learning prototype representations across few-shot tasks for event detection. In *Proceedings of the 2021 conference on empirical methods in natural language processing* (pp. 5270–5277). Online and Punta Cana, Dominican Republic: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.emnlp-main.427 doi: 10.18653/v1/2021.emnlp-main.427

Lai, V. D., Dernoncourt, F., & Nguyen, T. H. (2020). Exploiting the matching information in the support set for few shot event classification. In *Pakdd.*

Lai, V. D., Nguyen, M. V., Kaufman, H., & Nguyen, T. H. (2021, August). Event extraction from historical texts: A new dataset for black rebellions. In *Findings of the association for computational linguistics: Acl-ijcnlp 2021* (pp. 2390–2400). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.findings-acl.211 doi: 10.18653/v1/2021.findings-acl.211

Lai, V. D., Nguyen, M. V., Nguyen, T. H., & Dernoncourt, F. (2021). Graph learning regularization and transfer learning for few-shot event detection. In *Proceddings of the 44th international ACM SIGIR conference on research and development in information retrieval.*

Lai, V. D., & Nguyen, T. (2019, November). Extending event detection to new types with learning from keywords. In *Proceedings of the 5th workshop on noisy user-generated text (w-nut 2019)* (pp. 243–248). Hong Kong, China: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D19-5532 doi: 10.18653/v1/D19-5532

Lai, V. D., Nguyen, T. H., & Dernoncourt, F. (2020, July). Extensively matching for few-shot learning event detection. In *Proceedings of the first joint workshop on narrative understanding, storylines, and events* (pp. 38–45). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2020.nuse-1.5 doi: 10.18653/v1/2020.nuse-1.5

Lai, V. D., Nguyen, T. N., & Nguyen, T. H. (2020a, November). Event detection: Gate diversity and syntactic importance scores for graph convolution neural networks. In *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)* (pp. 5405–5411). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2020.emnlp-main.435 doi: 10.18653/v1/2020.emnlp-main.435

Lai, V. D., Nguyen, T. N., & Nguyen, T. H. (2020b, November). Event detection: Gate diversity and syntactic importance scores for graph convolution neural networks. In *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)* (pp. 5405–5411). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2020.emnlp-main.435 doi: 10.18653/v1/2020.emnlp-main.435

Lai, V. D., Veyseh, A. P. B., Nguyen, M. V., Dernoncourt, F., & Nguyen, T. H. (2022, October). MECI: A multilingual dataset for event causality identification. In *Proceedings of the 29th international conference on computational linguistics* (pp. 2346–2356). Gyeongju, Republic of Korea: International Committee on Computational Linguistics. Retrieved from https://aclanthology.org/2022.coling-1.206

Lample, G., Conneau, A., Denoyer, L., & Ranzato, M. (2017). Unsupervised machine translation using monolingual corpora only. *ICLR*.

LDC. (2005). *ACE (automatic content extraction) english annotation guidelines for events*. Linguistic Data Consortium. Retrieved from https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/english-events-guidelines-v5.4.3.pdf

Le, D., & Nguyen, T. H. (2021, April). Fine-grained event trigger detection. In *Proceedings of the 16th conference of the european chapter of the association for computational linguistics: Main volume* (pp. 2745–2752). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.eacl-main.237 doi: 10.18653/v1/2021.eacl-main.237

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

Lee, H., Recasens, M., Chang, A., Surdeanu, M., & Jurafsky, D. (2012, July). Joint entity and event coreference resolution across documents. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning* (pp. 489–500). Jeju Island, Korea: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D12-1045

Lee, K., Maji, S., Ravichandran, A., & Soatto, S. (2019). Meta-learning with differentiable convex optimization. In *Proceedings of the conference on Computer Vision and Pattern Recognition (CVPR).*

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., ... Zettlemoyer, L. (2020, July). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 7871–7880). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2020.acl-main.703 doi: 10.18653/v1/2020.acl-main.703

Li, D., Huang, L., Ji, H., & Han, J. (2019, June). Biomedical event extraction based on knowledge-driven tree-LSTM. In *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)* (pp. 1421–1430). Minneapolis, Minnesota: Association for Computational Linguistics. Retrieved from https://aclanthology.org/N19-1145 doi: 10.18653/v1/N19-1145

Li, F., Huang, R., Xiong, D., & Zhang, M. (2016, December). Learning event expressions via bilingual structure projection. In *Proceedings of COLING 2016, the 26th international conference on computational linguistics: Technical papers* (pp. 1441–1450). Osaka, Japan: The COLING 2016 Organizing Committee. Retrieved from https://aclanthology.org/C16-1136

Li, H., Ji, H., Deng, H., & Han, J. (2011). Exploiting background information networks to enhance bilingual event extraction through topic modeling. In *Proc. of international conference on advances in information mining and management.*

Li, L., Liu, Y., & Qin, M. (2018). Extracting biomedical events with parallel multi-pooling convolutional neural networks. *IEEE/ACM transactions on computational biology and bioinformatics*, *17*(2), 599–607.

Li, Q., Ji, H., & Huang, L. (2013, August). Joint event extraction via structured prediction with global features. In *Proceedings of the 51st annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 73–82). Sofia, Bulgaria: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/P13-1008`

Liao, S., & Grishman, R. (2010, July). Using document level cross-event inference to improve event extraction. In *Proceedings of the 48th annual meeting of the association for computational linguistics* (pp. 789–797). Uppsala, Sweden: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/P10-1081`

Liao, S., & Grishman, R. (2011, September). Acquiring topic features to improve event extraction: in pre-selected and balanced collections. In *Proceedings of the international conference recent advances in natural language processing 2011* (pp. 9–16). Hissar, Bulgaria: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/R11-1002`

Lin, Y., Ji, H., Huang, F., & Wu, L. (2020, July). A joint neural model for information extraction with global features. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 7999–8009). Online: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2020.acl-main.713` doi: 10.18653/v1/2020.acl-main.713

Liu, J., Chen, Y., Liu, K., Bi, W., & Liu, X. (2020, November). Event extraction as machine reading comprehension. In *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)* (pp. 1641–1651). Online: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2020.emnlp-main.128` doi: 10.18653/v1/2020.emnlp-main.128

Liu, J., Chen, Y., Liu, K., & Zhao, J. (2019, November). Neural cross-lingual event detection with minimal parallel resources. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)* (pp. 738–748). Hong Kong, China: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D19-1068 doi: 10.18653/v1/D19-1068

Liu, J., Chen, Y., & Zhao, J. (2021). Knowledge enhanced event causality identification with mention masking generalizations. In *Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence* (pp. 3608–3614). Retrieved from https://www.ijcai.org/proceedings/2020/0499.pdf

Liu, S., Chen, Y., He, S., Liu, K., & Zhao, J. (2016, August). Leveraging FrameNet to improve automatic event detection. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 2134–2143). Berlin, Germany: Association for Computational Linguistics. Retrieved from https://aclanthology.org/P16-1201 doi: 10.18653/v1/P16-1201

Liu, S., Chen, Y., Liu, K., & Zhao, J. (2017, July). Exploiting argument information to improve event detection via supervised attention mechanisms. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 1789–1798). Vancouver, Canada: Association for Computational Linguistics. Retrieved from https://aclanthology.org/P17-1164 doi: 10.18653/v1/P17-1164

Liu, X., Luo, Z., & Huang, H. (2018, October-November). Jointly multiple events extraction via attention-based graph information aggregation. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 1247–1256). Brussels, Belgium: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D18-1156 doi: 10.18653/v1/D18-1156

Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvininejad, M., ... Zettlemoyer, L. (2020). Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, *8*, 726–742. Retrieved from https://aclanthology.org/2020.tacl-1.47 doi: 10.1162/tacl_a_00343

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

Lou, C., Gao, J., Yu, C., Wang, W., Zhao, H., Tu, W., & Xu, R. (2022). Translation-based implicit annotation projection for zero-shot cross-lingual event argument extraction. In *Proceedings of the 45th international acm sigir conference on research and development in information retrieval* (pp. 2076–2081).

Lu, D., Subburathinam, A., Ji, H., May, J., Chang, S.-F., Sil, A., & Voss, C. (2020, May). Cross-lingual structure transfer for zero-resource event extraction. In *Proceedings of the twelfth language resources and evaluation conference* (pp. 1976–1981). Marseille, France: European Language Resources Association. Retrieved from https://aclanthology.org/2020.lrec-1.243

Lu, W., & Nguyen, T. H. (2018, October-November). Similar but not the same: Word sense disambiguation improves event detection via neural representation matching. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 4822–4828). Brussels, Belgium: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D18-1517 doi: 10.18653/v1/D18-1517

Lu, W., & Roth, D. (2012, July). Automatic event extraction with structured preference modeling. In *Proceedings of the 50th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 835–844). Jeju Island, Korea: Association for Computational Linguistics. Retrieved from https://aclanthology.org/P12-1088

Lu, Y., Lin, H., Xu, J., Han, X., Tang, J., Li, A., ... Chen, S. (2021, August). Text2Event: Controllable sequence-to-structure generation for end-to-end event extraction. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: Long papers)* (pp. 2795–2806). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.acl-long.217 doi: 10.18653/v1/2021.acl-long.217

Luan, Y., Wadden, D., He, L., Shah, A., Ostendorf, M., & Hajishirzi, H. (2019, June). A general framework for information extraction using dynamic span graphs. In *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)* (pp. 3036–3046). Minneapolis, Minnesota: Association for Computational Linguistics. Retrieved from https://aclanthology.org/N19-1308 doi: 10.18653/v1/N19-1308

Lyu, Q., Zhang, H., Sulem, E., & Roth, D. (2021, August). Zero-shot event extraction via transfer learning: Challenges and insights. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 2: Short papers)* (pp. 322–332). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.acl-short.42 doi: 10.18653/v1/2021.acl-short.42

Majumder, A., & Ekbal, A. (2015). Event extraction from biomedical text using crf and genetic algorithm. In *Proceedings of the 2015 third international conference on computer, communication, control and information technology (c3it)* (pp. 1–7).

Man, H., Nguyen, M., & Nguyen, T. (2022, July). Event causality identification via generation of important context words. In *Proceedings of the 11th joint conference on lexical and computational semantics* (pp. 323–330). Seattle, Washington: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2022.starsem-1.28 doi: 10.18653/v1/2022.starsem-1.28

Man Duc Trong, H., Trong Le, D., Pouran Ben Veyseh, A., Nguyen, T., & Nguyen, T. H. (2020, November). Introducing a new dataset for event detection in cybersecurity texts. In *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)* (pp. 5381–5390). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2020.emnlp-main.433 doi: 10.18653/v1/2020.emnlp-main.433

Marcheggiani, D., & Titov, I. (2017, September). Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 conference on empirical methods in natural language processing* (pp. 1506–1515). Copenhagen, Denmark: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D17-1159 doi: 10.18653/v1/D17-1159

McClosky, D., Surdeanu, M., & Manning, C. (2011, June). Event extraction as dependency parsing. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies* (pp. 1626–1635). Portland, Oregon, USA: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/P11-1163

M'hamdi, M., Freedman, M., & May, J. (2019, November). Contextualized cross-lingual event trigger extraction with minimal resources. In *Proceedings of the 23rd conference on computational natural language learning (conll)* (pp. 656–665). Hong Kong, China: Association for Computational Linguistics. Retrieved from https://aclanthology.org/K19-1061 doi: 10.18653/v1/K19-1061

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).

Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, *38*(11), 39–41.

Miller, G. A., Chodorow, M., Landes, S., Leacock, C., & Thomas, R. G. (1994). Using a semantic concordance for sense identification. In *Proceedings of the workshop on human language technology.*

Min, B., & Zhao, X. (2019, November). Measure country-level socio-economic indicators with streaming news: An empirical study. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)* (pp. 1249–1254). Hong Kong, China: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D19-1121 doi: 10.18653/v1/D19-1121

Minard, A.-L., Speranza, M., Agirre, E., Aldabe, I., van Erp, M., Magnini, B., ... Urizar, R. (2015, June). SemEval-2015 task 4: TimeLine: Cross-document event ordering. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)* (pp. 778–786). Denver, Colorado: Association for Computational Linguistics. Retrieved from https://aclanthology.org/S15-2132 doi: 10.18653/v1/S15-2132

Minh Tran, H., Phung, D., & Nguyen, T. H. (2021, August). Exploiting document structures and cluster consistencies for event coreference resolution. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: Long papers)* (pp. 4840–4850). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.acl-long.374 doi: 10.18653/v1/2021.acl-long.374

Mintz, M., Bills, S., Snow, R., & Jurafsky, D. (2009, August). Distant supervision for relation extraction without labeled data. In *Proceedings of the joint conference of the 47th annual meeting of the ACL and the 4th international joint conference on natural language processing of the AFNLP* (pp. 1003–1011). Suntec, Singapore: Association for Computational Linguistics. Retrieved from https://aclanthology.org/P09-1113

Mirza, P. (2014, June). Extracting temporal and causal relations between events. In *Proceedings of the ACL 2014 student research workshop* (pp. 10–17). Baltimore, Maryland, USA: Association for Computational Linguistics. Retrieved from https://aclanthology.org/P14-3002 doi: 10.3115/v1/P14-3002

Mirza, P., Sprugnoli, R., Tonelli, S., & Speranza, M. (2014, April). Annotating causality in the TempEval-3 corpus. In *Proceedings of the EACL 2014 workshop on computational approaches to causality in language (CAtoCL)* (pp. 10–19). Gothenburg, Sweden: Association for Computational Linguistics. Retrieved from https://aclanthology.org/W14-0702 doi: 10.3115/v1/W14-0702

Mirza, P., & Tonelli, S. (2014, August). An analysis of causality between events and its relation to temporal information. In *Proceedings of COLING 2014, the 25th international conference on computational linguistics: Technical papers* (pp. 2097–2106). Dublin, Ireland: Dublin City University and Association for Computational Linguistics. Retrieved from https://aclanthology.org/C14-1198

Mitamura, T., Liu, Z., & Hovy, E. (2015). Overview of TAC KBP 2015 event nugget track. In *TAC*.

Mitamura, T., Liu, Z., & Hovy, E. H. (2017). Events detection, coreference and sequencing: What's next? overview of the tac kbp 2017 event track. In *Tac*.

Mostafazadeh, N., Grealish, A., Chambers, N., Allen, J., & Vanderwende, L. (2016, June). CaTeRS: Causal and temporal relation scheme for semantic annotation of event structures. In *Proceedings of the fourth workshop on events* (pp. 51–61). San Diego, California: Association for Computational Linguistics. Retrieved from https://aclanthology.org/W16-1007 doi: 10.18653/v1/W16-1007

Neeleman, A., & Van de Koot, H. (2012). The linguistic expression of causation. *The theta system: Argument structure at the interface*, *20*.

Nguyen, M. V., Lai, V. D., & Nguyen, T. H. (2021, June). Cross-task instance representation interactions and label dependencies for joint information extraction with graph convolutional networks. In *Proceedings of the 2021 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 27–38). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.naacl-main.3 doi: 10.18653/v1/2021.naacl-main.3

Nguyen, M. V., Lai, V. D., Pouran Ben Veyseh, A., & Nguyen, T. H. (2021, April). Trankit: A light-weight transformer-based toolkit for multilingual natural language processing. In *Proceedings of the 16th conference of the european chapter of the association for computational linguistics: System demonstrations* (pp. 80–90). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.eacl-demos.10 doi: 10.18653/v1/2021.eacl-demos.10

Nguyen, M. V., Min, B., Dernoncourt, F., & Nguyen, T. (2022, July). Joint extraction of entities, relations, and events via modeling inter-instance and inter-label dependencies. In *Proceedings of the 2022 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 4363–4374). Seattle, United States: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2022.naacl-main.324 doi: 10.18653/v1/2022.naacl-main.324

Nguyen, M. V., & Nguyen, T. H. (2021, April). Improving cross-lingual transfer for event argument extraction with language-universal sentence structures. In *Proceedings of the sixth arabic natural language processing workshop* (pp. 237–243). Kyiv, Ukraine (Virtual): Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.wanlp-1.27

Nguyen, T. H., Cho, K., & Grishman, R. (2016, June). Joint event extraction via recurrent neural networks. In *Proceedings of the 2016 conference of the north American chapter of the association for computational linguistics: Human language technologies* (pp. 300–309). San Diego, California: Association for Computational Linguistics. Retrieved from https://aclanthology.org/N16-1034 doi: 10.18653/v1/N16-1034

Nguyen, T. H., Fu, L., Cho, K., & Grishman, R. (2016, August). A two-stage approach for extending event detection to new types via neural networks. In *Proceedings of the 1st workshop on representation learning for NLP* (pp. 158–165). Berlin, Germany: Association for Computational Linguistics. Retrieved from https://aclanthology.org/W16-1618 doi: 10.18653/v1/W16-1618

Nguyen, T. H., & Grishman, R. (2015, July). Event detection and domain adaptation with convolutional neural networks. In *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)* (pp. 365–371). Beijing, China: Association for Computational Linguistics. Retrieved from https://aclanthology.org/P15-2060 doi: 10.3115/v1/P15-2060

Nguyen, T. H., & Grishman, R. (2016, November). Modeling skip-grams for event detection with convolutional neural networks. In *Proceedings of the 2016 conference on empirical methods in natural language processing* (pp. 886–891). Austin, Texas: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D16-1085 doi: 10.18653/v1/D16-1085

Nguyen, T. H., & Grishman, R. (2018). Graph convolutional networks with argument-aware pooling for event detection. In *Thirty-second aaai conference on artificial intelligence.* Retrieved from https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16329/16155

Nguyen, T. H., Meyers, A., & Grishman, R. (2016g). New york university 2016 system for kbp event nugget: A deep learning approach. In *Proceedings of text analysis conference (tac).*

Nguyen, T. M., & Nguyen, T. H. (2019). One for all: Neural joint modeling of entities and events. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 33, pp. 6851–6858).

Ning, Q., Feng, Z., & Roth, D. (2017, September). A structured learning approach to temporal relation extraction. In *Proceedings of the 2017 conference on empirical methods in natural language processing* (pp. 1027–1037). Copenhagen, Denmark: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D17-1108 doi: 10.18653/v1/D17-1108

Ning, Q., Feng, Z., Wu, H., & Roth, D. (2018, July). Joint reasoning for temporal and causal relations. In *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 2278–2288). Melbourne, Australia: Association for Computational Linguistics. Retrieved from https://aclanthology.org/P18-1212 doi: 10.18653/v1/P18-1212

Nivre, J., de Marneffe, M.-C., Ginter, F., Goldberg, Y., Hajič, J., Manning, C. D., ... Zeman, D. (2016, May). Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the tenth international conference on language resources and evaluation (LREC'16)* (pp. 1659–1666). Portorož, Slovenia: European Language Resources Association (ELRA). Retrieved from https://aclanthology.org/L16-1262

O'Gorman, T., Wright-Bettner, K., & Palmer, M. (2016, November). Richer event description: Integrating event coreference with temporal, causal and bridging annotation. In *Proceedings of the 2nd workshop on computing news storylines (CNS 2016)* (pp. 47–56). Austin, Texas: Association for Computational Linguistics. Retrieved from https://aclanthology.org/W16-5706 doi: 10.18653/v1/W16-5706

Oh, J.-H., Torisawa, K., Hashimoto, C., Iida, R., Tanaka, M., & Kloetzer, J. (2016). A semi-supervised learning approach to why-question answering. In *Thirtieth aaai conference on artificial intelligence.*

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., ... Ray, A. (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, *35*, 27730–27744.

Patwardhan, S., & Riloff, E. (2009, August). A unified model of phrasal and sentential evidence for information extraction. In *Proceedings of the 2009 conference on empirical methods in natural language processing* (pp. 151–160). Singapore: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D09-1016

Peng, H., Song, Y., & Roth, D. (2016, November). Event detection and co-reference with minimal supervision. In *Proceedings of the 2016 conference on empirical methods in natural language processing* (pp. 392–402). Austin, Texas: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D16-1038 doi: 10.18653/v1/D16-1038

Pennington, J., Socher, R., & Manning, C. (2014, October). GloVe: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532–1543). Doha, Qatar: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D14-1162 doi: 10.3115/v1/D14-1162

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018, June). Deep contextualized word representations. In *Proceedings of the 2018 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 1 (long papers)* (pp. 2227–2237). New Orleans, Louisiana: Association for Computational Linguistics. Retrieved from https://aclanthology.org/N18-1202 doi: 10.18653/v1/N18-1202

Peyre, G., & Cuturi, M. (2019). Computational optimal transport: With applications to data science. In *Foundations and trends in machine learning.*

Phung, D., Minh Tran, H., Nguyen, M. V., & Nguyen, T. H. (2021, November). Learning cross-lingual representations for event coreference resolution with multi-view alignment and optimal transport. In *Proceedings of the 1st workshop on multilingual representation learning* (pp. 62–73). Punta Cana, Dominican Republic: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.mrl-1.6 doi: 10.18653/v1/2021.mrl-1.6

Piskorski, J., Belayeva, J., & Atkinson, M. (2011, September). Exploring the usefulness of cross-lingual information fusion for refining real-time news event extraction: A preliminary study. In *Proceedings of the international conference recent advances in natural language processing 2011* (pp. 210–217). Hissar, Bulgaria: Association for Computational Linguistics. Retrieved from https://aclanthology.org/R11-1029

Pouran Ben Veyseh, A., & Nguyen, T. (2022, July). Word-label alignment for event detection: A new perspective via optimal transport. In *Proceedings of the 11th joint conference on lexical and computational semantics* (pp. 132–138). Seattle, Washington: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2022.starsem-1.11 doi: 10.18653/v1/2022.starsem-1.11

Pouran Ben Veyseh, A., Nguyen, T. H., & Dou, D. (2019, July). Graph based neural networks for event factuality prediction using syntactic and semantic structures. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 4393–4399). Florence, Italy: Association for Computational Linguistics. Retrieved from https://aclanthology.org/P19-1432 doi: 10.18653/v1/P19-1432

Prasad, R., Dinesh, N., Lee, A., Miltsakaki, E., Robaldo, L., Joshi, A., & Webber, B. (2008). The penn discourse treebank 2.0. In *Proceedings of the sixth international conference on language resources and evaluation (lrec'08).*

Press, O., Smith, N., & Lewis, M. (2021). Train short, test long: Attention with linear biases enables input length extrapolation. In *International conference on learning representations.*

Pustejovsky, J., Castaño, J. M., Ingria, R., Saurí, R., Gaizauskas, R. J., Setzer, A., ... Radev, D. R. (2003). Timeml: Robust specification of event and temporal expressions in text. In *New directions in question answering.*

Pustejovsky, J., Hanks, P., Sauri, R., See, A., Gaizauskas, R., Setzer, A., ... Ferro, L. (2003). The timebank corpus. In *Corpus linguistics* (Vol. 2003, p. 40).

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, *1*(8), 9.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, *21*(140), 1–67. Retrieved from http://jmlr.org/papers/v21/20-074.html

Rahimi, Z., & Shamsfard, M. (2021). Persian causality corpus (percause) and the causality detection benchmark. *CoRR*, *abs/2106.14165*. Retrieved from https://arxiv.org/abs/2106.14165

Rehbein, I., & Ruppenhofer, J. (2020, May). A new resource for German causal language. In *Proceedings of the twelfth language resources and evaluation conference* (pp. 5968–5977). Marseille, France: European Language Resources Association. Retrieved from https://aclanthology.org/2020.lrec-1.731

Ruder, S., Søgaard, A., & Vulić, I. (2019, July). Unsupervised cross-lingual representation learning. In *Proceedings of the 57th annual meeting of the association for computational linguistics: Tutorial abstracts* (pp. 31–38). Florence, Italy: Association for Computational Linguistics. Retrieved from https://aclanthology.org/P19-4007 doi: 10.18653/v1/P19-4007

Sadek, J., & Meziane, F. (2018). Building a causation annotated corpus: the salford arabic causal bank-proclitics. In *Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018).* European Language Resources Association (ELRA). Retrieved from http://lrec-conf.org/workshops/lrec2018/W30/pdf/11_W30.pdf

Sahu, S. K., Christopoulou, F., Miwa, M., & Ananiadou, S. (2019, July). Inter-sentence relation extraction with document-level graph convolutional neural network. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 4309–4316). Florence, Italy: Association for Computational Linguistics. Retrieved from https://aclanthology.org/P19-1423 doi: 10.18653/v1/P19-1423

Satyapanich, T., Ferraro, F., & Finin, T. (2020). Casie: Extracting cybersecurity event information from text. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 34, pp. 8749–8757). Retrieved from https://ojs.aaai.org/index.php/AAAI/article/view/6401/6257

Schweter, S. (2020, April). *Berturk - bert models for turkish.* Zenodo. Retrieved from https://doi.org/10.5281/zenodo.3770924 doi: 10.5281/zenodo.3770924

Sha, L., Qian, F., Chang, B., & Sui, Z. (2018). Jointly extracting event triggers and arguments by dependency-bridge rnn and tensor-based argument interaction. In *Thirty-second aaai conference on artificial intelligence.* Retrieved from https://shalei120.github.io/docs/sha2018Joint.pdf

Shahaf, D., & Guestrin, C. (2010). Connecting the dots between news articles. In *Proceedings of the 16th acm sigkdd international conference on knowledge discovery and data mining* (p. 623–632). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/1835804.1835884 doi: 10.1145/1835804.1835884

Shalyminov, I., Lee, S., Eshghi, A., & Lemon, O. (2019). Few-shot dialogue generation without annotated data: A transfer learning approach. In *Proceedings of the 20th annual sigdial meeting on discourse and dialogue.*

Shen, S., Wu, T., Qi, G., Li, Y.-F., Haffari, G., & Bi, S. (2021, August). Adaptive knowledge-enhanced Bayesian meta-learning for few-shot event detection. In *Findings of the association for computational linguistics: Acl-ijcnlp 2021* (pp. 2417–2429). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.findings-acl.214 doi: 10.18653/v1/2021.findings-acl.214

Sims, M., Park, J. H., & Bamman, D. (2019, July). Literary event detection. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 3623–3634). Florence, Italy: Association for Computational Linguistics. Retrieved from https://aclanthology.org/P19-1353 doi: 10.18653/v1/P19-1353

Snell, J., Swersky, K., & Zemel, R. (2017). Prototypical networks for few-shot learning. *Advances in neural information processing systems*, *30*.

Speer, R., Chin, J., & Havasi, C. (2017). Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-first aaai conference on artificial intelligence.*

Sprugnoli, R., & Tonelli, S. (2019, June). Novel event detection and classification for historical texts. *Computational Linguistics*, *45*(2), 229–265. Retrieved from https://aclanthology.org/J19-2002 doi: 10.1162/coli_a_00347

Steen, J., & Markert, K. (2019, November). Abstractive timeline summarization. In *Proceedings of the 2nd workshop on new frontiers in summarization* (pp. 21–31). Hong Kong, China: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D19-5403 doi: 10.18653/v1/D19-5403

Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., & Tsujii, J. (2012, April). brat: a web-based tool for NLP-assisted text annotation. In *Proceedings of the demonstrations at the 13th conference of the European chapter of the association for computational linguistics* (pp. 102–107). Avignon, France: Association for Computational Linguistics. Retrieved from https://aclanthology.org/E12-2021

Subburathinam, A., Lu, D., Ji, H., May, J., Chang, S.-F., Sil, A., & Voss, C. (2019, November). Cross-lingual structure transfer for relation and event extraction. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)* (pp. 313–325). Hong Kong, China: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D19-1030 doi: 10.18653/v1/D19-1030

Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H., & Hospedales, T. M. (2018). Learning to compare: Relation network for few-shot learning. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 1199–1208).

Tong, M., Xu, B., Wang, S., Cao, Y., Hou, L., Li, J., & Xie, J. (2020, July). Improving event detection via open-domain trigger knowledge. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 5887–5897). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2020.acl-main.522 doi: 10.18653/v1/2020.acl-main.522

Tran Phu, M., Nguyen, M. V., & Nguyen, T. H. (2021, November). Fine-grained temporal relation extraction with ordered-neuron LSTM and graph convolutional networks. In *Proceedings of the seventh workshop on noisy user-generated text (w-nut 2021)* (pp. 35–45). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.wnut-1.5 doi: 10.18653/v1/2021.wnut-1.5

Tran Phu, M., & Nguyen, T. H. (2021, June). Graph convolutional networks for event causality identification with rich document-level structures. In *Proceedings of the 2021 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 3480–3490). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.naacl-main.273 doi: 10.18653/v1/2021.naacl-main.273

Trong, H. M. D., Ngo, N. T., Ngo, L. V., & Nguyen, T. H. (2022). Selecting optimal context sentences for event-event relation extraction. In *Proceedings of the association for the advancement of artificial intelligence (aaai).*

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2018). Graph attention networks. *ICLR*.

Venugopal, D., Chen, C., Gogate, V., & Ng, V. (2014, October). Relieving the computational bottleneck: Joint inference for event extraction with high-dimensional features. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 831–843). Doha, Qatar: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D14-1090 doi: 10.3115/v1/D14-1090

Verhagen, M., Saurí, R., Caselli, T., & Pustejovsky, J. (2010, July). SemEval-2010 task 13: TempEval-2. In *Proceedings of the 5th international workshop on semantic evaluation* (pp. 57–62). Uppsala, Sweden: Association for Computational Linguistics. Retrieved from https://aclanthology.org/S10-1010

Veyseh, A. P. B., Lai, V., Dernoncourt, F., & Nguyen, T. H. (2021, August). Unleash GPT-2 power for event detection. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: Long papers)* (pp. 6271–6282). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.acl-long.490 doi: 10.18653/v1/2021.acl-long.490

Veyseh, A. P. B., Nguyen, M. V., Dernoncourt, F., & Nguyen, T. (2022, July). MINION: a large-scale and diverse dataset for multilingual event detection. In *Proceedings of the 2022 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 2286–2299). Seattle, United States: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2022.naacl-main.166 doi: 10.18653/v1/2022.naacl-main.166

Veyseh, A. P. B., Nguyen, M. V., Ngo, N. T., Min, B., & Nguyen, T. H. (2021, November). Modeling document-level context for event detection via important context selection. In *Proceedings of the 2021 conference on empirical methods in natural language processing* (pp. 5403–5413). Online and Punta Cana, Dominican Republic: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.emnlp-main.439 doi: 10.18653/v1/2021.emnlp-main.439

Vinyals, O., Blundell, C., Lillicrap, T., & Wierstra, D. (2016). Matching networks for one shot learning. *Advances in neural information processing systems*, *29*.

Wadden, D., Wennberg, U., Luan, Y., & Hajishirzi, H. (2019, November). Entity, relation, and event extraction with contextualized span representations. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)* (pp. 5784–5789). Hong Kong, China: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D19-1585 doi: 10.18653/v1/D19-1585

Walker, C., Strassel, S., Medero, J., & Maeda, K. (2006). Ace 2005 multilingual training corpus. In *Technical report, linguistic data consortium.*

Wang, H., Chen, M., Zhang, H., & Roth, D. (2020, November). Joint constrained learning for event-event relation extraction. In *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)* (pp. 696–706). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2020.emnlp-main.51 doi: 10.18653/v1/2020.emnlp-main.51

Wang, H., Gan, Z., Liu, X., Liu, J., Gao, J., & Wang, H. (2019, November). Adversarial domain adaptation for machine reading comprehension. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)* (pp. 2510–2520). Hong Kong, China: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D19-1254 doi: 10.18653/v1/D19-1254

Wang, H., Zhang, H., Chen, M., & Roth, D. (2021, November). Learning constraints and descriptive segmentation for subevent detection. In *Proceedings of the 2021 conference on empirical methods in natural language processing* (pp. 5216–5226). Online and Punta Cana, Dominican Republic: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.emnlp-main.423 doi: 10.18653/v1/2021.emnlp-main.423

Wang, X., Jia, S., Han, X., Liu, Z., Li, J., Li, P., & Zhou, J. (2020, December). Neural Gibbs Sampling for Joint Event Argument Extraction. In *Proceedings of the 1st conference of the asia-pacific chapter of the association for computational linguistics and the 10th international joint conference on natural language processing* (pp. 169–180). Suzhou, China: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2020.aacl-main.21

Wang, X., Wang, Z., Han, X., Jiang, W., Han, R., Liu, Z., ... Zhou, J. (2020, November). MAVEN: A Massive General Domain Event Detection Dataset. In *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)* (pp. 1652–1671). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2020.emnlp-main.129 doi: 10.18653/v1/2020.emnlp-main.129

Webber, B., Prasad, R., Lee, A., & Joshi, A. (2019). The penn discourse treebank 3.0 annotation manual. *Philadelphia, University of Pennsylvania*.

Wolff, P. (2007). Representing causation. *Journal of experimental psychology: General*, *136*(1), 82.

Wu, X., Huang, K.-H., Fung, Y., & Ji, H. (2022, July). Cross-document misinformation detection based on event graph reasoning. In *Proceedings of the 2022 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 543–558). Seattle, United States: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2022.naacl-main.40 doi: 10.18653/v1/2022.naacl-main.40

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., ... Macherey, K. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Xiong, C., Merity, S., & Socher, R. (2016). Dynamic memory networks for visual and textual question answering. In *Proceedings of the international conference on machine learning (icml).*

Xu, R., Liu, T., Li, L., & Chang, B. (2021, August). Document-level event extraction via heterogeneous graph-based interaction model with a tracker. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: Long papers)* (pp. 3533–3546). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.acl-long.274 doi: 10.18653/v1/2021.acl-long.274

Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., ... Raffel, C. (2021, June). mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 483–498). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.naacl-main.41 doi: 10.18653/v1/2021.naacl-main.41

Yan, H., Jin, X., Meng, X., Guo, J., & Cheng, X. (2019, November). Event detection with multi-order graph convolution and aggregated attention. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)* (pp. 5766–5770). Hong Kong, China: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D19-1582 doi: 10.18653/v1/D19-1582

Yang, H., Chen, Y., Liu, K., Xiao, Y., & Zhao, J. (2018, July). DCFEE: A document-level Chinese financial event extraction system based on automatically labeled training data. In *Proceedings of ACL 2018, system demonstrations* (pp. 50–55). Melbourne, Australia: Association for Computational Linguistics. Retrieved from https://aclanthology.org/P18-4009 doi: 10.18653/v1/P18-4009

Yang, S., Feng, D., Qiao, L., Kan, Z., & Li, D. (2019, July). Exploring pre-trained language models for event extraction and generation. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 5284–5294). Florence, Italy: Association for Computational Linguistics. Retrieved from https://aclanthology.org/P19-1522 doi: 10.18653/v1/P19-1522

Yao, W., Dai, Z., Ramaswamy, M., Min, B., & Huang, R. (2020, November). Weakly Supervised Subevent Knowledge Acquisition. In *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)* (pp. 5345–5356). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2020.emnlp-main.430 doi: 10.18653/v1/2020.emnlp-main.430

Zeng, Y., Feng, Y., Ma, R., Wang, Z., Yan, R., Shi, C., & Zhao, D. (2018). Scale up event extraction learning via automatic training data generation. In *Thirty-second aaai conference on artificial intelligence.*

Zhang, H., Wang, H., & Roth, D. (2021, August). Zero-shot Label-aware Event Trigger and Argument Classification. In *Findings of the association for computational linguistics: Acl-ijcnlp 2021* (pp. 1331–1340). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.findings-acl.114 doi: 10.18653/v1/2021.findings-acl.114

Zhang, Y., Qi, P., & Manning, C. D. (2018, October-November). Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 2205–2215). Brussels, Belgium: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D18-1244 doi: 10.18653/v1/D18-1244

Zhang, Z., & Ji, H. (2021, June). Abstract Meaning Representation guided graph encoding and decoding for joint information extraction. In *Proceedings of the 2021 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 39–49). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.naacl-main.4 doi: 10.18653/v1/2021.naacl-main.4

Zhang, Z., Xu, W., & Chen, Q. (2016). Joint event extraction based on skip-window convolutional neural networks. In *Natural language understanding and intelligent applications* (pp. 324–334). Springer.

Zhao, Y., Jin, X., Wang, Y., & Cheng, X. (2018, July). Document embedding enhanced event detection with hierarchical and supervised attention. In *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 2: Short papers)* (pp. 414–419). Melbourne, Australia: Association for Computational Linguistics. Retrieved from https://aclanthology.org/P18-2066 doi: 10.18653/v1/P18-2066

Zheng, S., Cao, W., Xu, W., & Bian, J. (2019, November). Doc2EDAG: An end-to-end document-level framework for Chinese financial event extraction. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)* (pp. 337–346). Hong Kong, China: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D19-1032 doi: 10.18653/v1/D19-1032

Zhou, B., Ning, Q., Khashabi, D., & Roth, D. (2020, July). Temporal common sense acquisition with minimal supervision. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 7579–7589). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2020.acl-main.678 doi: 10.18653/v1/2020.acl-main.678

Zuo, X., Cao, P., Chen, Y., Liu, K., Zhao, J., Peng, W., & Chen, Y. (2021a, August). Improving event causality identification via self-supervised representation learning on external causal statement. In *Findings of the association for computational linguistics: Acl-ijcnlp 2021* (pp. 2162–2172). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.findings-acl.190 doi: 10.18653/v1/2021.findings-acl.190

Zuo, X., Cao, P., Chen, Y., Liu, K., Zhao, J., Peng, W., & Chen, Y. (2021b, August). LearnDA: Learnable knowledge-guided data augmentation for event causality identification. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: Long papers)* (pp. 3558–3571). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.acl-long.276 doi: 10.18653/v1/2021.acl-long.276

Zuo, X., Chen, Y., Liu, K., & Zhao, J. (2020, December). KnowDis: Knowledge enhanced data augmentation for event causality detection via distant supervision. In *Proceedings of the 28th international conference on computational linguistics* (pp. 1544–1550). Barcelona, Spain (Online): International Committee on Computational Linguistics. Retrieved from https://aclanthology.org/2020.coling-main.135 doi: 10.18653/v1/2020.coling-main.135