STRUCTURE-BASED MODELS FOR NEURAL INFORMATION EXTRACTION

by

AMIR POURAN BEN VEYSEH

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

June 2023

DISSERTATION APPROVAL PAGE

Student: Amir Pouran Ben Veyseh

Title: Structure-based Models for Neural Information 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
Humphrey Shi	Core Member
Thanh Nguyen	Core Member
Kristopher Kyle	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 June 2023

DISSERTATION ABSTRACT

Amir Pouran Ben Veyseh

Doctor of Philosophy

Department of Computer Science

June 2023

Title: Structure-based Models for Neural Information Extraction

Information Extraction (IE) is one of the important fields in Natural Language Processing. IE models can be exploited to obtain meaningful information from raw text and provide them in a structured format which can be used for downstream applications such as question answering. An IE system consists of several tasks including entity recognition, relation extraction, and event detection, to name a few. Among all recent advanced deep learning models proposed for IE tasks, one of the potential directions to improve performance is to incorporate structural information. Structural information refers to encoding any interactions between different parts of the input text. This information is helpful to overcome long distances between related words or sentences. In this dissertation, we study novel deep learning models that integrate structural information into the representation learning process. In particular, three major categories, i.e., existing structures, inferred structure at the sample level, and inferred structure at dataset levels are studied in this dissertation. We finally showcase the novel application of structure-based models for the less-explored setting of cross-lingual IE.

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

CURRICULUM VITAE

NAME OF AUTHOR: Amir Pouran Ben Veyseh

GRADUATE AND UNDERGRADUATE SCHOOLS ATTENDED:

University of Oregon, Eugene, OR, USA
University of Tehran, Tehran, Iran
Amirkabir University of Technology, Tehran, Iran

DEGREES AWARDED:

Master of Science, Computer Engineering, 2018, University of Tehran
Bachelor of Science, Computer Engineering, 2014, Amirkabir University of Technology

AREAS OF SPECIAL INTEREST:

Deep Learning
Natural Language Processing
Information Extraction

PROFESSIONAL EXPERIENCE:

Graduate Research Assistant, University of Oregon, 2019-2023
Graduate Teaching Assistant, University of Oregon, 2018-2019
Research Intern, Adobe Research, 2019-2021

GRANTS, AWARDS AND HONORS:

Gurdeep Pall Scholarship Award, University of Oregon, 2021
Adobe Fellowship Award, Adobe Inc., 2021
Gurdeep Pall Scholarship Award, University of Oregon, 2019
Best Paper Award, EACL Conference, 2021

PUBLICATIONS:

Veyseh, A., Nguyen, M., Dernoncourt, F., Bonan, M., Nguyen, T. *"Document-Level Event Argument Extraction via Optimal Transport"* ACL, 2022

Veyseh, A., Xu, N., Tran, Q., Manjunatha, V., Dernoncourt, F., Nguyen, T. *"Transfer Learning and Prediction Consistency for Detecting Offensive Spans of Text"* ACL, 2022

Veyseh, A., Nguyen, M., Trung, N., Min, B., Nguyen, T. *"Modeling Document-Level Context for Event Detection via Important Context Selection"* EMNLP, 2021

Veyseh, A., Nguyen, M., Min, B., Nguyen, T. *"Augmenting Open-Domain Event Detection with Synthetic Data from GPT-2"* ECML-PKDD, 2021

Veyseh, A., Dernoncourt, F., Nguyen, T., Chang, W., Celi, L., *"Acronym Identification and Disambiguation Shared Tasks for Scientific Document Understanding"* Workshop on Scientific Document Understanding at AAAI, 2021

Veyseh, A., Lai, V., Dernoncourt, F., Nguyen, T. *"Unleash GPT-2 Power for Event Detection"* ACL, 2021

Veyseh, A., Dernoncourt, F., Nguyen, T. *"DPR at SemEval-2021 Task 8: Dynamic Path Reasoning for Measurement Relation Extraction"* SemEval@ACL-IJCNLP, 2021

Veyseh, A., Dernoncourt, F., Chang, W., Nguyen, T. *"MadDog: A Web-based System for Acronym Identification and Disambiguation"* EACL, System Demonstrations, 2021 (**Best Paper Award**)

Nguyen, M., Lai, V., Veyseh, A., Nguyen, T. *"Trankit: A Light-Weight Transformer-based Toolkit for Multilingual Natural Language Processing"* EACL, System Demonstrations, 2021 (**Outstanding Paper Award**)

Veyseh, A., Dernoncourt, F., Tran, Q., Manjunatha, V., Wang, L., Jain, R., Kim, D., Chang, W., Nguyen, T. *"Inducing Rich Interaction Structures between Words for Document-level Event Argument Extraction"* PAKDD 2021

Veyseh, A., Dernoncourt, F., Tran, Q., Nguyen, T. *"What Does This Acronym Mean? Introducing a New Dataset for Acronym Identification and Disambiguation"* COLING 2020

Veyseh, A., Nguyen, T., Nguyen, T. *"Graph Transformer Networks with Syntactic and Semantic Structures for Event Argument Extraction"*. EMNLP Findings, 2020

Trong, H., Le, D., Veyseh, A., Nguyen, T., Nguyen, T. *"Introducing a New Dataset for Event Detection in Cybersecurity Texts"*. EMNLP, 2020

Veyseh, A., Nouri, N., Dernoncourt, F., Dou, D., Nguyen, T. *"Introducing Syntactic Structures into Target Opinion Word Extraction with Deep Learning"*. EMNLP, 2020

Veyseh, A., Dernoncourt, F., Nguyen, T. *"Improving Slot Filling by Utilizing Contextual Information"*. NLP4ConvAI workshop at ACL, 2020

Veyseh, A., Dernoncourt, F., Dou, D., Nguyen, T. *"Exploiting the Syntax-Model Consistency for Neural Relation Extraction"*. ACL, 2020

Veyseh, A., Nouri, N., Dernoncourt, F., Tran, Q., Dou, D., Nguyen, T. *"Improving Aspect-based Sentiment Analysis using Gated Graph Convolution Network and Syntax-based Regulation"*. EMNLP Findings, 2020

Veyseh, A., Dernoncourt, F., Thai, M., Dou, D., Nguyen, T. *"Multi-view Consistency for Relation Extraction via Mutual Information and Structure Prediction"*. AAAI, 2020

Veyseh, A., Dernoncourt, F., Dou, D., Nguyen, T. *"A Joint Model for Definition Extraction with Syntactic Connection and Semantic Consistency"*. AAAI, 2020

Veyseh, A., Thai, M., Nguyen, T., Dou, D. *"Rumor Detection in Social Networks via*
Deep Contextual Modeling". ASONAM, 2019

Veyseh, A., Nguyen, T., Dou, D. *"Improving Cross-Domain Performance for Relation Extraction via Dependency Prediction and Information Flow Control"*. IJCAI, 2019

Veyseh, A., Nguyen, T., Dou, D. *"Graph based Neural Networks for Event Factuality Prediction using Syntactic and Semantic Structures"*. ACL, 2019

Veyseh, A., Ebrahimi, J., Dou, D., Lowd, D. *"Temporal Attentional Model for Rumor Stance Classification"*. CIKM, 2017

Veyseh, A. *"Cross-lingual Question Answering Using Common Semantic Space"*. Workshop on Graph-based Methods for Natural Language Processing, NAACL-HLT 2016.

ACKNOWLEDGEMENTS

I greatly thank my advisor Prof. Thien Huu Nguyen for all of his support and advice during my Ph.D. Under his valuable supervision, I was able to build my research experience and expand my connections with the research community.

I also appreciate all supports that I received from Dr. Franck Dernoncourt during my internships at Adobe Research. His mentorship was integral in creating a productive research experience at Adobe for me.

Finally, I'd like to give gratitude to all faculty members of my dissertation committee members and all members of Dr. Nguyen's research group for all their constructive feedback during my progress on this dissertation.

TABLE OF CONTENTS

Chapter Page

LIST OF FIGURES

Figure Page

LIST OF TABLES

Table Page

Table Page

Table Page

CHAPTER I

INTRODUCTION

Information Extraction (IE) is one of the important fields of natural language processing (NLP) with the primary goal of creating structured knowledge from unstructured text. In more than two decades, IE has gained a lot of attention and many new tasks and models have been proposed. Moreover, with the proliferation of deep learning and neural nets in recent years, the advanced deep models have brought about a surge in the performance of IE models. Among others, some of the existing deep models resort to structure-based modeling whose goal is to exploit the structure of the text (i.e., interactions of different parts of the text) or external structures (e.g., a knowledge base). In this Chapter, we will review the structure-based deep models proposed for various IE tasks and also other related NLP tasks. Finally, we will discuss the limitations of the existing models and the potential for future work. [1]

1.1 Structural Data for Information Extraction

Textual materials such as books and websites are still one of the major sources of information in human societies. In the Big Data era and with the expansion of the world wide web and social networks in recent years, the amount of available textual data has also increased substantially. While on the one hand the sheer size of these resources provides valuable information about many topics, on the other hand, it hinders efficient information look-up. To address this limitation, one possible solution is to store the information in pre-defined structures (i.e., knowledge bases) so it can be quickly retrieved. Since converting the information

[1]The next Chapters of this dissertation contain materials from published and co-authored papers. We acknowledge all co-authors, Franck Dernoncourt, Quan Tran, Varun Manjunatha, Lidan Wang, Rajiv Jain, Doo Soon Kim, Walter Chang, My Thai, Dejing Dou, Javid Ebrahimi, and Thien Huu Nguyen.

available in textual resources into structured knowledge bases is tedious and the KB could quickly get obsolete, automatic approaches to extract structured information from free text is necessary. These automatic approaches are called Information Extraction (IE) methods and consist of several tasks including: 1) Identifying the real-world entities (e.g., person, company, and dates) that have been mentioned in text (i.e., Named Entity Recognition) Lample, Ballesteros, Subramanian, Kawakami, and Dyer (2016); Mikheev, Moens, and Grover (1999); Nadeau and Sekine (2007), 2) Assigning unique identity (e.g., entity IDs in a knowledge base) to the entity mentions in text (i.e., Entity Linking) Hachey, Radford, Nothman, Honnibal, and Curran (2013); T. Lin, Etzioni, et al. (2012); X. Liu et al. (2013), 3) Finding all expressions (e.g., proper nouns and pronouns) that refer to the same entity (i.e., Co-reference Resolution) Lee, He, Lewis, and Zettlemoyer (2017); Ng and Cardie (2002); Raghunathan et al. (2010) 4) Detecting the semantic relationships between entities that are specified in text (e.g., ownership and marriage) (i.e., Relation Extraction) Y. Lin, Shen, Liu, Luan, and Sun (2016); Mintz, Bills, Snow, and Jurafsky (2009); Zelenko, Aone, and Richardella (2003) and 5) Finding information about incidents referred to in text (e.g., divorce and attack); this information might answer questions like "*who did what to whom?*" (i.e., Event Extraction) Ahn (2006); T. H. Nguyen, Cho, and Grishman (2016); Ritter, Etzioni, and Clark (2012).

In more than the last two decades, extensive research has been conducted to design effective methods for each of the aforementioned IE tasks. These techniques range from rule-based methods Eftimov, Koroušić Seljak, and Korošec (2017), to feature-based models G. Zhou, Su, Zhang, and Zhang (2005a) and recent advanced deep learning models Rao, Marcu, Knight, and Daumé III (2017). As it has been

shown in other NLP tasks including text summarization Mani, Bloedorn, and Gates (1998), document classification Y. Zhang, Yu, et al. (2020), question answering Qiu et al. (2019) and machine translation Ma, Tamura, Utiyama, Sumita, and Zhao (2019), incorporating the existing structures into deep models for IE could improve their performance. The employed structure could either refer to syntactic structure, e.g., dependency tree Bunescu and Mooney (2005), semantic structure, e.g., entity similarity graph Min, Shi, Grishman, and Lin (2012), or external structures (e.g., knowledge base) Z. Fang et al. (2020). In this Chapter, we study techniques that employ structure-based modeling to improve performance on various IE tasks. In addition, we review the application of text structure in other related NLP tasks. Finally, we discuss their limitations and the possible directions for future work.

1.2 Named Entity Recognition

Named entity recognition (NER) is the first task in the information extraction pipeline and it aims to identify words or phrases that refer to people, organizations, locations, etc. This task has been extensively studied in the more than last two decades. Approaches for this task extend from the unsupervised rule-based methods Collins and Singer (1999), to the supervised feature engineering G. Zhou and Su (2002) and the advanced deep learning models Dernoncourt, Lee, and Szolovits (2017). Two sub-tasks for this problem should be solved:

- Named entity recognition: The first step for NER is to identify the sub-sequences of the input text that refer to real-world entities. For instance, in the input text *Kabul is controlled by President Abdol Mosharaf's government, which Taleban is fighting to overthrow*, the model should identify the phrases *Kabul*, *Abdol Mosharaf* and *Taleban* as the named entity mentions.

– Named entity classification: The next step for NER is to classify the recognized named entity mentions to one of the pre-defined types. For instance, in the aforementioned example, the model should be able to classify *Kabul* as *Location*, *Abdol Mosharaf* as *Person* and *Taleban* as *Miscellaneous*.

Most of the existing models address the two tasks simultaneously. However, some of the prior work proposes a different model for each task. For instance, Collins and Singer (1999) introduced a new rule-based model to predict the named entity type using the spelling of the name and the context in which it appears. The spelling rule might use some look-up tables or predefined patterns (e.g., the existence of *Mr* indicates the type *Person*). On the other hand, the contextual rules could refer to dependencies between a type and some indicative words in the surroundings of the named entity (e.g., *president* in the above example). Similar rules have been used in a subsequent work G. Zhou and Su (2002), however, in G. Zhou and Su (2002) authors employed Hidden Markov Model (HMM) to simultaneously identify the named entity mentions and their types. The HMM model is able to consider the previous tags and also the features of the current word to predict its label.

While the feature-based models have gained some improvements on NER, the state-of-the-art models are now employing deep learning models. NER models can benefit from the pre-trained word embeddings T. H. Nguyen, Sil, Dinu, and Florian (2016) and the non-linearity of the deep learning-based models J. Li, Sun, Han, and Li (2020). More interestingly, these models could also incorporate structural information resulting in better performance on NER Aguilar and Solorio (2019); Jie and Lu (2019); J. Yu, Bohnet, and Poesio (2020). The structure could refer to the syntactic parse tree. In Jie and Lu (2019), authors show that a deep

model enhanced with the dependency tree could have two advantages: 1) The syntactic connection between the words is indicative of the entity type, 2) the long-dependencies captured from the dependency tree could improve the representation of the words for the NER task. In this work, the authors proposed an LSTM-CRF model to capture this information. More specifically, the representation of the words to the LSTM model is enhanced with the representation of their parents and the dependency relation between them. Afterward, the interaction between the parent and its child is models via an interaction function (e.g., dot product or a feed-forward neural net) over the corresponding hidden states of the parent and children from the LSTM layer. In another recent work Aguilar and Solorio (2019), authors propose to encode the syntactic structure using Tree-LSTM. Furthermore, they introduce local and global attention. The local attention highlights important words with respect to the current word that is being evaluated. On the other hand, global attention emphasizes the important words of the sentence without restricting the attention to the current word.

1.3 Entity Linking

Entity linking (EL) is the task of identifying the corresponding entities in a knowledge base (KB) for every entity mention in the text. For instance, in the given example *Michael Jordan has recently signed a new contract with his new club. He will be the first goalkeeper of the Rangers for two years.*, there are two entity mentions: 1) Michael Jordan and 2) Rangers. Both of these entity mentions could refer to multiple entities (e.g., Michael Jordan could refer to the English goalkeeper, American football offensive lineman or American former professional basketball player and the Rangers could refer to the entities Rangers football club in South Africa, an association football club from Glasgow or Rangers football club in New

Zealand). The correct mapping between the entity mentions and the entities in the KB depends on the context and the relation between entities in the KB.

As KBs are structured, this task inherently benefits from encoding the structure (here, structure mainly refers to the relationships between entities in the KB). While several feature-based models have been proposed for EL Ji and Grishman (2011); Khalife and Vazirgiannis (2018); Veyseh (2016), the state-of-the-art performances are achieved using deep learning models Z. Fang et al. (2019); L. Wu, Petroni, Josifoski, Riedel, and Zettlemoyer (2019); Yamada and Shindo (2019). Most of the existing work breaks down EL into two sub-tasks Sevgili, Shelmanov, Arkhipov, Panchenko, and Biemann (2020):

- Candidate Generation: Using string similarity or descriptions available in KB, a list of candidate entities is generated Sevgili et al. (2020). For instance, for the given example, the model compares the similarity between the entity mention *Rangers* and the entities in the KB to extract the list of Rangers football club in South Africa, an association football club from Glasgow or Rangers football club in New Zealand. Authors in P. Le and Titov (2019); Zwicklbauer, Seifert, and Granitzer (2016) use a simple string-match to find the list of entities. Others might use the aliases for the KB entities computed from the knowledge base metadata (e.g., redirect pages in Wikipedia) Z. Fang et al. (2019) or pre-calculated prior probabilities (e.g., computed from mention-link count statistics) Ganea and Hofmann (2017)

- Entity Ranking: Based on the consistency between the context of the entity mention and the representations of the entities, the candidate entities are ranked to choose the entity with the highest score Sevgili et al. (2020). For instance, in the given example, the model will choose the entity *an*

association football club from Glasgow as the most likely entity among the extracted list of possible entities for the entity mention *Rangers*

For the second sub-task, to represent the entities and encode their similarities with the entity mentions in the text, KB structure and the relationship between different entity mentions in the text could be helpful and the recent work has shown that deep graph architectures are able to efficiently encode this information Z. Fang et al. (2020); J. Wu et al. (2020). One of the early works that applied graph convolution network (GCN) for entity linking is Cao, Hou, Li, and Liu (2018). They employed GCN to model the coherency between the candidate entities. In order to handle the large number of entities in the KB, they proposed to apply GCN only on the subset of entities extracted in the first phase (i.e., candidate generation). In another work, authors in Z. Fang et al. (2020) proposed a graph attention network to attend to the previous and next entity mentions in the text to encode the sequential inter-dependencies between the entity mentions in the text. Authors in J. Wu et al. (2020) propose to dynamically compute and refine the graph structure to model the dependencies between entities. The dynamic graph computation has been shown to be effective for other related tasks too Nan, Guo, Sekulic, and Lu (2020). In this method, the representation of the nodes is used to compute the structure of the graph for the next iteration of the graph convolution network.

1.4 Coreference Resolution

Coreference resolution (CR) is a fundamental task of IE whose goal is to identify different entity mentions in the document that refer to the same entity. For instance, in the example *"I voted for Nader because he was most aligned with my values," Sara said*, there are three entity mentions for the person *Sara* (i.e., *I*,

my and *Sara*) and two entity mentions for the person *Nader* (i.e., *Nader* and *he*). A CR model should be able to find the chain of entity mentions for the entities *Sara* and *Nader*.This task is crucial for many downstream applications including Relating Extraction and Question Answering.

According to Stylianou and Vlahavas (2019), traditional methods for CR can be categorized into four categories:

- Mention-pair: This method determines if a pair of mentions refer to the same thing. This method employs the features of the two mentions and performs a binary classification Soon, Ng, and Lim (2001).

- Mention-ranking: In this category, models collectively consider all mentions to resolve a specific mention. More specifically, for each mention, all candidate antecedents are ranked and the one with the highest score is selected to be chained to the current mention Rahman and Ng (2009).

- Entity-based methods: Models of this category employ clustering techniques to decide if two clusters of mentions should be merged or not Ratinov and Roth (2012).

- Latent structure models: These models create a hierarchy of the mentions to collectively cluster them Björkelund and Kuhn (2014). The major difference between entity-based and latent structure models is that, contrary to the former which employs agglomerative clustering, in the latter, the clusters are created in a tree-like structure.

Similar techniques have been also employed in deep learning models Clark and Manning (2016); Lee, He, and Zettlemoyer (2018); Wiseman, Rush, Shieber,

and Weston (2015). In addition, some deep learning models formulate this task as question answering W. Wu, Wang, Yuan, Wu, and Li (2020) or they use reinforcement learning to perform this task Fei, Li, Li, and Li (2019). While the traditional methods have proven the importance of text structure (i.e., dependency tree) for this task Björkelund and Kuhn (2014); Lappin and Leass (1994), only recently syntactical structure has been used in deep models K. Fang and Jian (2019). Authors in K. Fang and Jian (2019) proposed to use the syntactic structure of the sentence for Chinese coreference resolution. The syntactic structure has three purposes in this work: (1) To filter out unlikely entity mentions. More specifically, they keep only those candidate entity mentions (i.e., spans of words) that are represented by a node in the syntactic tree; (2) To represent the context. In particular, the syntactic tree traverse is employed to gather the syntax-based context for each entity mention (i.e. node in the syntactic tree); (3) Encode structural features (e.g., degree of the node or its siblings).

1.5 Relation Extraction

Relation extraction (RE) is the task of identifying the semantic relation between entity mentions in the text. For instance, in the given example *Some Arab countries also want to play a role in the stable operation in Iraq but are reluctant to send troops because of political, religious and ethnic considerations, the official said*, a relationship of *Organization-Affiliation* is mentioned between entities *Arab countries* and *troops*. An RE model should be able to extract the relationship between different entity mentions or decide whether or not the entities of interest are in a relation.

This task has been extensively studied and several settings for that have been proposed including single-sentence, document-level, distantly supervised,

end-to-end, and cross-domain. In this survey, we first review the most important existing works and datasets for each of these settings of RE. Afterward, we provide details on the existing structure-aware deep RE models.

1.5.1 Single-sentence. In this setting, the input to the model will be only one sentence consisting of at least two entity mentions. The goal is to predict the relation type between every pair of entity mentions in the input sentence. For this setting, the major existing datasets include ACE Doddington et al. (2004), TACRED Y. Zhang, Zhong, Chen, Angeli, and Manning (2017) and SemEval 2010 Task 8 Hendrickx et al. (2009). The ACE dataset is a series of datasets, i.e. ACE 2003, ACE 2004, ACE 2005, ACE 2007, and ACE 2008, released by NIST for the entity, relation, and event extraction. The SemEval 2010 Task 8 dataset provides 8,853 instances for 9 relation types. The relations in the SemEval dataset is directed meaning that the total number of relations will be 18 plus one special relation (i.e., *Other*) for entities that are not in a relation. The corpus to be annotated for the SemEval dataset is obtained via a pattern-based search for each relation type from the Web. Despite the vast application of these two datasets for sentence-level relation extraction, there are at least two limitations in them. First, these datasets cover a limited number of relation types (i.e., 19 relations in SemEval and 24 relations in ACE 2003 and 2004 datasets). This small number of relations will not represent the challenges in a real-world application of RE. Second, the common issue in both ACE and SemEval datasets is that these datasets are relatively small for data-hungry deep learning models. In other words, this small size prevents the models from more effective feature extractions from the data. To address this limitation, authors in Y. Zhang et al. (2017) proposed a new large-scaled sentence-level relation extraction dataset, i.e., TACRED. This

dataset contains 106,264 examples (both positive (i.e., examples in which the two entities are in a relation) and negative (i.e., examples in which the entity are not in a relation)) in 42 relation types. The annotation is conducted over the TAC KBP evaluations from 2009 to 2015. The annotation refers to the relations between organizations, people, and locations.

The traditional methods for sentence-level relation extraction use feature-based and statistical models Bunescu and Mooney (2005); Chan and Roth (2010); Sun, Grishman, and Sekine (2011); G. Zhou, Su, Zhang, and Zhang (2005b). The major limitation of the feature-based models is that it requires extensive feature engineering efforts and domain knowledge to find the effective patterns for the relation mentions. Moreover, these models cannot generalize well to unseen data. To address these limitations, deep learning models are employed for RE and they have gained considerable attention from the community M. Nguyen and Nguyen (2018b); T. H. Nguyen and Grishman (2015c, 2015a, 2016); L. Wang, Cao, de Melo, and Liu (2016); D. Zeng, Liu, Lai, Zhou, and Zhao (2014); Y. Zhang et al. (2017); P. Zhou et al. (2016). Using deep architectures, e.g., Convolutional Neural Net (CNN) and Long Short-Term Memory (LSTM), along with the background knowledge provided via word embeddings, deep models reached the state-of-the-art performance on different datasets. In addition, some of the deep models embrace the findings of the feature-based models to improve RE performance. For instance, using dependency trees in deep learning models has been shown to be effective for deep learning-based RE models. Y. Liu et al. (2015); Miwa and Bansal (2016); Xu et al. (2015a); Y. Zhang, Qi, and Manning (2018). For this purpose, graph neural networks (GNN) could be employed to model the dependency structure. Y. Zhang et al. (2018) proposed one of the early GNN-based models for RE. One of

the major challenges to employ the dependency tree in a deep model is that neural models operating directly on parse trees are usually difficult to parallelize and thus computationally inefficient Y. Zhang et al. (2018). To address this issue, the prior work pruned the dependency tree to keep only the words on the shortest dependency path (SDP) between the two entity mentions in the dependency tree. However, such simplification will result in loss of information as some of the words off the path could be also important. To address this issue, Y. Zhang et al. (2018) proposed to use graph convolution networks (GCN) Kipf and Welling (2016). GCNs are able to efficiently encode the graph structures with the parameter sharing. In order to improve the performance, Y. Zhang et al. (2018) also proposed to prune the dependency tree along with the SDP up to a pre-defined distance between the off-the-path and on-the-path words. Their evaluations of TACRED dataset prove the effectiveness of this method.

1.5.2 Document-level. In this category of RE models, the input to the system is a document consisting of multiple entities. Entity mentions might appear in one sentence or across multiple sentences in the given document. In general, the relation mentions in documents could be categorized into two groups: 1) Intra-sentence relations: If both entity mentions that are in relation are mentioned in the same sentence, the relation between them is an intra-sentence relation; 2) Inter-sentence relations: In this category, the two entity mentions appear in different sentences across the document. For instantce, in the given document *Elias Brown (May 9, 1793– July 7, 1857) was a U.S. Representative from Maryland. Born near Baltimore, Maryland, Brown attended the common schools. He died near Baltimore, Maryland, and is interred in a private cemetery near Eldersburg, Maryland.*, the relation between the entity *U.S.* and *Maryland*

is *COUNTRY* and the relation between the entity *Maryland* and *Baltimore* is *LOCATED_IN*. As both relations can be inferred from the immediate sentence in which the entities appear, the two mentioned relations are intra-sentence relations. On the other hand, the relation between the entity *Baltimore* and *U.S.* is *COUNTRY* that should be inferred from the different sentences in which the entity mentions appear. Thus, this relation is of type inter-sentence relations.

While there are some domain-specific J. Li et al. (2016) or distantly supervised N. Peng, Poon, Quirk, Toutanova, and Yih (2017); Quirk and Poon (2016) document-level relation extraction datasets, the only large scale manually labeled document-level relation extraction dataset available is provided by Yao et al. (2019). This dataset, called DocRED, contains 56,354 relation facts and 132,357 entity annotations across 5,053 Wikipedia documents. Among all relation facts, 40.7% of them are inter-sentence relations which require inference in document level.

The major challenge for document level relation extraction is to infer the long range dependencies between the entities across sentences. To deal with this issue, most of the existing work propose to employ structure-based modeling. More specifically, a structure that could represent the dependencies between different parts of the document is constructed, either using some heuristics S. Zeng, Xu, Chang, and Li (2020) or it is learned by a trainable component Nan et al. (2020). In order to infer a task specific structure for document-level RE, authors in Christopoulou, Miwa, and Ananiadou (2019) proposed to infer the document structure from the representations of its edges. More specifically, they first create a dense graph whose vertices are the entity mentions, sentences and the entities (i.e., the people, organizations, etc that have been mentioned in

the document). The entity mentions are represented using their corresponding hidden states of a bi-directional LSTM (BiLSTM) network. The sentence and the entity representations are computed by pooling the representations of all words or mentions of them, respectively. Afterwards, the representations of edges of the graph are obtained using the representation of their heads and tails. Finally, to compute the representations for longer paths (e.g., paths consisting of two edges), a feed forward neural net is employed to combine the representations of all edges in that path. The path representation between the two entity mentions of interest is used to predict the relation. While this work proposed a method to infer the structure-based entity mentions relations, it fails to dynamically update the representations of the nodes, including the entity mentions themselves. To address this issue, authors in Nan et al. (2020) proposed a structure inference mechanism to dynamically and consecutively update the node representation and the graph structure, in turn. More specifically, after obtaining the representations of the nodes[2], the weights of all edges in the dense graph are computed from the head and tail representation of the edge. Afterwards, a GCN layer is employed to update the node representations. Using the updated representation of the nodes, a new set of weights for edges of the graph is computed. This process is repeated for N times. Finally, the representation of the two entity mentions are used for relation prediction.

Most recently, authors in D. Wang, Hu, Cao, and Sun (2020) proposed another saturate-based document-level relation extraction model. In the proposed approach, authors first construct a set of nodes based on the sentences, entities, and mentions. Afterwards, similar to prior work, they connect the nodes based

[2]in this work, entity mentions, the words on the SDP between every pair of entity mentions and the entities themselves serve as the nodes of the graph

on some heuristics (e..g, if a mention is hosted by a sentence there would be connection between the corresponding sentence node and the mention node). Using the obtained global graph and a GCN model, authors update the initial representations of the nodes which are obtained from a sequence-based encoder. In the next step, the representations of the nodes are updated using multi-head self-attention component. This component could capture the semantic dependencies between the extracted nodes, i.e., sentences, mentions and entities. Finally, by concatenating the representations obtained from the GCN layer and the self-attention layer for the two entities of interest, the final representation vector is constructed and it is consumed by a logistic regression classifier to predict the semantic relations between the two entity mentions in the document.

1.5.3 Distantly Supervised. One of the major challenges for RE is that collecting training data is expensive. Thus, the exiting datasets are quite small, specifically for data-hungry deep models. One remedy to this issue could be to use distantly supervised (DS) datasets. In this setting, some heuristics are employed to collect examples for pre-defined relation sets. In the seminal work Mintz et al. (2009), authors employed the relations between entities in Freebase knowledge base and an unlabeled corpus to extract examples for each relation. For instance, consider the two entities *Steve Jobs* and *Apple*. Suppose that the relation between this two entity mention in the KB is *Works_At*. Based on the method proposed by Mintz et al. (2009), one could extract examples for relation *Works_At* by extracting all sentences in a large corpus (e.g., Wikipedia) that contains mentions for both entities *Steve Jobs* and *Apple*

While the distantly supervised RE dataset could extend the size of training sets, they also introduce noisy examples. More specifically, sentences that contain

both entities of interest but do not mention the supposed relation between entities are incorrectly labeled. This examples are indeed the false positives. Due to this problem, a distantly supervised RE model should be able to deal with the noisy example which might be extracted in this process. To this end, several techniques are proposed to exclude or rectify the incorrectly labeled samples in the training data. Some of the prior works exploit reinforcement learning (RL) to identify the incorrectly labeled samples. Feng, Huang, Zhao, Yang, and Zhu (2018) introduced a two-module RE model. The first module is an instance selector which identifies the instances with incorrect labels and filter them out. The second module is a relation classification model which use the input training data to learn the RE task. The reward for the instance selector is computed using the performance of the second component on the evaluation set. In a similar approach, Qin, Xu, and Wang Qin, Xu, and Wang (2018) proposed to employ RL to denoise the training data. However, in their method, instead of excluding the noisy samples, they suggested to change the label of the false positives to *None*, indicating there is no relation between the two entity mentions in the sentence.

One issue with the RL-based approaches is that they make a hard decision to either exclude or change the label of noisy samples. In other words, during the training of the relation classifier, the hard labels of the noisy samples might be detrimental for the training process. In order to alleviate the effect of the incorrect hard labels, T. Liu, Wang, Chang, and Sui (2017) introduced a soft-label multi-instance learning method for relation extraction with noisy training samples. In this method, all samples of a pair of entity mentions h_i and t_j are grouped into the set $< h_i, t_j >$ consisting of c sentences $S = \{x_1, x_2, \ldots, x_c\}$. The set $< h_i, t_j >$ could be represented either by only one of the sentences in S or an attention-based

pooling of the sentences. Afterwards, to obtain the label for the set $< h_i, h_t >$, instead of using the one-hot $L_{i,j}$ vector label obtained from the distantly supervised dataset, they proposed to learn a dense vector $\bar{L}_{i,j}$ from the bag representation and the one-hot vector $L_{i,j}$. The soft label $\bar{L}_{i,j}$ will be used in the next epoch by the relation classifier as the gold label for the set $< h_i, t_j >$.

For the distantly supervised relation extraction setting, the structure-based modeling has been also shown to be effective. The graph-based models employed for DS relation extraction encode the structure of the knowledge graph. More specifically, the structure of the knowledge base is employed to model the interaction between the entities, thereby, denoise the samples for every pair of entity mentions. For instance, authors in Hu et al. (2019) proposed to employ the knowledge graph structure to learn an embedding vector for each relation type. More specifically, using the graph encoding method proposed by Bordes, Usunier, Garcia-Duran, Weston, and Yakhnenko (2013), they learn the representation of the head (h), tail (t) and relation (r) of the triples $< h, r, t >$ in the knowledge graph. Afterwards, using the representation of the relations in the training set, an attention score is computed for each sentence in the training set. The attention-based representation of the sentences are employed by the relation classifier.

In addition to the application of the graph structure for denoising the samples in DS datasets, some researchers have employed graph structure to learn the dependencies between relations predicted for an entity pair (e_1, e_2) from a set of sentences $S = \{s_1, s_2, \ldots, s_n\}$. Two relation are dependent on each other, if the existence of one infer the existence of another. Note that it would be a directed dependency. For instance, *President_of* between a person and a country could also induce the relation *Lives_in* between the person and the country. To encode this

dependency, authors in Shang, Huang, Sun, and Mao (2020) proposed to build a graph structure where the nodes are the relation types and the edges could represent the dependencies between them. During training the model is optimized to learn a dependency relation graph for every pair of entities that could represent the gold relations between the two entity.

1.5.4 End-to-end. Relation extraction is the task of identifying the relations between entity mentions in text. To this end, the entity mentions should be first identified. While a pipeline approach identifies the entities and relations in separate stages, the major limitation is that the errors in the entity recognition stage could be propagated to the relation extraction stage. In order to prevent this error propagation, an end-to-end (E2E) RE model jointly recognizes the entity mentions and the the relation between them in a given text snippet.

While the sentence-level relation extraction datasets (e.g., ACE or SemEval 2010 Task 8) could be used to train and evaluate an E2E RE model Miwa and Bansal (2016), for this setting, most of the recent work report the performance of the models on NYT Miwa and Bansal (2016) and WebNLG Gardent, Shimorina, Narayan, and Perez-Beltrachini (2017) datasets. NYT was originally proposed to address the high level of noise in the datasets prepared by the distant supervision technique Mintz et al. (2009). To this end, they proposed a semi-supervised method to extract relation tipples (i.e., ($entity_1$,$relation$,$entity_2$)) from New York Times using the Freebase as the knowledge base. WebNLG is a corpus created using a natural language generation (NLG) framework operated on the DBpedia knowledge base.

Although prior work for E2E RE employed feature-engineering methods T.-V. T. Nguyen and Moschitti (2011), recent deep models are proved to achieve

the state-of-the-art results for this task Miwa and Bansal (2016). Moreover, in the recent work T.-J. Fu, Li, and Ma (2019), authors have shown that the structure-based modeling could improve the performance of an E2E RE system. In particular, two graph structures are employed in this work: (1) The syntactic tree of the sentence is employed by a graph convolution network (GCN) to enrich word representations The syntax-enriched word representations are employed to predict the entities and also the relation types between words; (2) A full-graph consisting of the words as the nodes and the pair-wise relations between words as the edges is created. In this graph, the edges (i.e., relations) that are predicted in the first stage (i.e., using the dependency based GCN) are emphasized by giving more attention weights to them. The main purpose of this graph is to encode the relation dependencies between words. Specifically, for those relations that share an entity (e.g., the head entity), the dependency between them could be encoded by the GCN layer to infer the direct relation between the other entities (e.g. the tail entities). For instance, if the triples *(BarackObama,LiveIn, WhiteHouse)* and *(WhiteHouse, PresidentialPalace, UnitedStates)* are predicted in the first stage, the second stage employ GCN to infer the third triple *(BarackObama, PresidentOf, UnitedStates)*. In addition, in order to predict multiple relations between every pair of entities, authors proposed to use a threshold in which every relation type r between the pair of words w_1 and w_2 (i.e., two predicted entity mentions), predicted in the second phase, will be included in the final model's prediction to create the triple (w_1, r, w_2).

1.5.5 Cross-domain. While the aforementioned settings suppose that the training and the evaluation data come from the same domain, it could not be guaranteed in all scenarios. For those cases that the RE model is trained and

evaluated in different domains, a cross-domain RE system is required. The main challenge of such a setting is that the features that are useful during training might not be relevant or helpful in the evaluation phase. To train and evaluate models in this setting, the ACE 2005 datasets is widely used. In this dataset, there are 6 different domains, i.e., (*bc, bn, cts, nw, un,* and *wl*), covering text from news, conversations and web blogs. Cross-domain models are trained on one of these domains (e.g., news) and are evaluated on the other domains (e..g, conversations and web blogs). Similar to the other settings, for cross-domain RE, prior work started to employ feature-based models M. Yu, Gormley, and Dredze (2015). However, deep models are proved to be more effective for this setting T. H. Nguyen and Grishman (2016). Until recently, the graph-based deep model have not been explored for this task. Recently, Veyseh, Nguyen, and Dou (2019) have shown that the structure of the text (e.g., dependency tree) could be used to improve the performance for cross-domain RE. Also, in the recent work Veyseh, Dernoncourt, Thai, Dou, and Nguyen (2020), they have employed deep learning to infer the structure of the text without using off-the-shelf parsers. More specifically, they propose to employ two deep architectures, i.e., ordered-neuron LSTM Shen, Tan, Sordoni, and Courville (2018) and self-attention mechanism Vaswani et al. (2017a), to infer two views of structure of the input sentence. Afterwards, by exploiting a neural-based mutual information estimator Belghazi et al. (2018), they increased the consistency between two structural views. Their evaluation on ACE 2005 dataset show that this techniques achieves the state-of-the-art results for cross-domain relation extraction.

1.6 Event Extraction

Event extraction is the task of identifying real word incidents mentioned in text such as *attack*, *divorce*, or *birth*. According to the ACE annotation guidelines, an event is described as something that happens and change the state of an entity. For instance, the sentence *Ames recruited her as an informant in 1983, then married her two years later*, implies that the marriage status of *Ames* is changed so it refers to an event of *marriage*. According to ACE annotation guidelines, every event mention consist of two components:

- Trigger: This is the word or phrase which most clearly express the occurrence of the event. It could be a verb, noun or adjective. For instance, in the sentence *John robert bond was born in England*, the verb *born* is the event trigger which indicates the occurance of the event *BE-BORN*. Note that each event trigger evokes a specific incident known as event type.

- Argument: Those entities that are participants of the event and their states are changed due to the occurrence of the event are considered as the event argument. In addition to the event participants, the other attributes of the event, e.g., time or location of the event, are also considered as the event arguments. For instance, in the sentence *The man accused of killing seven people near Boston on Tuesday got his guns in Massachusetts*, there is an event mention of *Kill*. The trigger word for this event is *killing* and the arguments of this event are *man*, *seven people*, *Boston* and *Tuesday*. It is worth noting that each argument takes a specific role in the event. For instance, in the given example, the role of the argument *seven people* is *victim* and the role of the argument *Boston* is *place*.

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

Table 1. Event Types and Sub-Types in ACE 2005 Xiang and Wang (2019)

The task of identifying the trigger and its type is known as Event Detection (ED) and the task of identifying the event arguments and their roles is known as Event Argument Extraction (EAE). For each of this tasks there is a wealth of prior work extending from feature-based models Ahn (2006); Hong et al. (2011a); Ji and Grishman (2008); Q. Li, Ji, and Huang (2013); Liao and Grishman (2010a, 2010b); McClosky, Surdeanu, and Manning (2011); Miwa, Thompson, Korkontzelos, and Ananiadou (2014); Patwardhan and Riloff (2009); Riedel and McCallum (2011); B. Yang and Mitchell (2016a) to advanced deep learning systems Y. Chen, Xu, Liu, Zeng, and Zhao (2015); T. M. Nguyen and Nguyen (2019a); Sha, Qian, Chang, and Sui (2018); S. Yang, Feng, Qiao, Kan, and Li (2019); J. Zhang, Qin, Zhang, Liu, and Ji (2019); Y. Zhang, Xu, et al. (2020). While most of the prior work consider sentence-level event extraction, some recent work has also introduced event extraction in document level Ebner, Xia, Culkin, Rawlins, and Van Durme (2019). Moreover, in addition to the text-based event extraction, there are some recent work that attempt to extract event mentions from multiple modalities (e.g., text and image) M. Li, Zareian, et al. (2020); T. Zhang et al. (2017). Furthermore, some prior work consider the open event extraction which aims to extract the event triggers without the assumption of a pre-defined domain (i.e., ontology of event

Domain	Proportion
Newswire	20%
Broadcast News	20%
Broadcast Conversations	15%
Weblog	15%
Usenet News Group	15%
Conversational Telephone Speech	15%

Table 2. Domain Statistics of the English portion of the ACE 2005 Xiang and Wang (2019)

types) Naik and Rosé (2020); Sims, Park, and Bamman (2019a); R. Wang, Zhou, and He (2019). Event extraction systems could be employed in knowledge base construction, question answering, and text summarization. In this section, we will review the important existing work and their major advantages and limitations. In the reviews of the models, we emphasize the application of text structure for event detection and event argument extraction.

1.6.1 Datasets. The most popular dataset among event extraction researches is ACE 2005 dataset. It has annotations for 599 documents with 6,000 labels for events Xiang and Wang (2019). The events are annotated with 8 types and 33 sub-types. Table 24 shows the event types and sub-types in ACE 2005 dataset. Docuemnts annotated for ACE 2005 are in English, Arabic and Chinese from six different domains, i.e., Newswire, Broadcast News, Broadcast Conversations, Weblog, Usenet News Group, and Conversational Telephone Speech. Table 2 shows the statistics of each of these domains in English section of ACE 2005 dataset.

In addition to ACE 2005 dataset, there are other datasets that are exploited by event extraction works:

	CASIE	CySecED
# event types	5	30
# positive examples	8,470	8,014
# negative examples	240,682	282,220
# sentences per document (average)	16.69	24.94

Table 3. Statistics for CASIE and CySecED. Negative examples refer to non-trigger words while positive examples are the annotated trigger words for the event types of interest Hieu Man Duc Trong (2020).

- TAC-KBP: introduced by Linguistic Data Consortium (LDC) "Rich Ere Annotation Guidelines Overview" (2016), provides annotations for 360 documents with 9 event types and 38 event sub-types.

- LitBank: This dataset annotates 100 English literary texts. It includes annotations for both entities and event triggers. Unlike ACE, LitBanck does not provide event types for the triggers.

- TimeBank: This dataset, provided by LDC Pustejovsky et al. (2003), includes annotations for events, times, and temporal relation between event mentions. Similar to LitBank, this dataset also does not provide types of the event triggers.

- Domain-Specific Datasets: In addition to the general-domain event annotation, some datasets focus on domain-specific datasets. BioNLP-ST is a collection of event mention annotations from various corpora including GENIA event corpus, BioInfer corpus, Gene regulation event corpus, GeneReg corpus and PPI corpora Nédellec et al. (2013); Vanegas, Matos, González, and Oliveira (2015); Xiang and Wang (2019). Another domain that has gained attention for event extraction is cyber-security domain. In this domain, event are categorized into four general topics: (1) Discover: Events referring to

identification of a vulnerability in a system, (2) Patch: Events mentioning the fixes of a known vulnerability, (3) Attack: Exploitation of a vulnerability to impact the system and (4) Impact: consequences of an attack on a system Hieu Man Duc Trong (2020). For cyber-security domain, CySecED Hieu Man Duc Trong (2020) and CASIE Satyapanich, Ferraro, and Finin (2020a) are the largest datasets available. The statistics of these datasets are provided in Table 3.

– Multi-modal Event Extraction: In additon to text-based event extraction, some recent work proposed a new dataset for extracting events from both textual and visual data M. Li, Zareian, et al. (2020).

1.6.2 Feature-based Models. Early work on event extraction has employed feature engineering for event extraction from text. In the early stages of event extraction research, Riloff and Shoen Riloff and Shoen (1995), proposed a pattern-based EE system. In their system, the syntactic parse of the sentence is employed to extract general patterns for event mentions. For instance, in the sentence *World trade center was bombed by terrorists*, identifying the subject (i.e., *Wordl trade center*), verb phrase (i.e., *was bombed*) and prepositional phrase (i.e., *by terrorists*) could lead to the event patterns *[x] was bombed* and *bombed by [y]* to identify the attack event and its arguments in text Xiang and Wang (2019). Based on the statistics of the patterns in the corpus, the high confident patterns are selected to be used in evaluation phase. Later in the following years, feature-based models employed statistical models such as nearest neighbors Ahn (2006), maximum-entropy learner Z. Chen and Ji (2009b), support vector machine Saha, Majumder, Hasanuzzaman, and Ekbal (2011), and conditional random field Majumder and Ekbal (2015).These models employ the lexical forms of the words,

the syntactic parse (e.g., the POS tag, the parent or children of the word in the dependency tree, or the label of the dependency edges), synonyms of the words, and the event or entity type Xiang and Wang (2019). For a complete review of these methods, refer to the survey provided by Xiang and Wang (2019).

1.6.3 Deep Models. Despite all progress obtained from more effective features employed in statistical models, the major limitations of feature-based systems is that these models are not able to incorporate background knowledge and also to infer new useful patterns from the training data. Deep learning addresses these limitations by utilizing the word embeddings pre-trained on large corpus and also by exploiting deep architectures to induce effective patterns from the training data. Due to these advantages, the recent event extraction systems employ deep learning Y. Chen et al. (2015); T. M. Nguyen and Nguyen (2019a); Sha et al. (2018); S. Yang et al. (2019); J. Zhang et al. (2019); Y. Zhang, Xu, et al. (2020). Some of the deep models exploit sequence-based architectures Sha et al. (2018), convolutional neural networks (CNN) Björne and Salakoski (2018), or recent transformer-based models Ahmad, Peng, and Chang (2020).

In addition to the deep architectures and background knowledge, some recent models attempted to incorporate the interaction between event types W. Li, Cheng, He, Wang, and Jin (2019) or argument roles X. Wang, Wang, et al. (2019) using hierarchy-based modeling. For instance, authors in X. Wang, Wang, et al. (2019), proposed to encode the hierarchy of event argument types using neural module network (NMN) Andreas, Rohrbach, Darrell, and Klein (2016). In particular, the hierarchy of the event argument types are employed to capture the dependency between related argument roles. For instance, in the sentence *Steve Jobs sold Pixar to Disney in 2006*, identifying the role of the entity *Steve Jobs* as

Seller and its hierarchical dependency with role *Buyer* (i.e., considering the fact that both *Seller* and *Buyer* are entities of type *Person* or *Organization*) could help the model to predict the role of the entity *disney* as *Buyer*. To encode this hierarchical information, authors proposed to train separate attention functions for each type which could be applied to the input sentence to obtain type-dependent repression of the input text. The aggregation of type-dependent representations of all possible types of an entity is used in the final classifier to predict the role of the entity.

1.6.4 Graph-based Models. The structure-based modeling has two applications for event extraction: 1) Text Representation and 2) Event Graph. In this section, we study each of them in details

1.6.4.1 Text Representation. The syntax or semantic based structures of the sentence might be employed by deep models to encode the interactions between the words, thereby improving the performance of event detection or event argument extraction. For instance, authors in Amir Pouran Ben Veyseh (2020) proposed to infer the task-specific syntactic and semantic structure of the input sentence using deep architectures. Specifically, the syntactic structure is induced by feeding the pair of dependency-based distance of the words to the trigger/argument into a feed forward neural net. The output of the feed forward neural net are employed as the entries of the syntax-based adjacency matrix. To infer the semantic structure of the input text, authors propose to employ self-attention mechanism Vaswani et al. (2017a). Finally, for efficient combination of the syntactic and semantic structures, graph transformer network (GTN) Yun, Jeong, Kim, Kang, and Kim (2019) is employed. GTN uses convolution operation to combine the structures and also encode the heterogeneous

paths by multiplying the adjacency matrix of all structures. One limitation of the GTN architecture is that it could result in overfitting to the training data due to the increased number of parameters for combining the structures. In order to alleviate this issue, authors in Amir Pouran Ben Veyseh (2020) proposed to employ information bottleneck technique. Specifically, they decrease the mutual information between the input and output of the GTN, treating this network as information bottleneck. This technique could prevents the model from memorizing patterns specific to the training data.

In another work, authors in D. Li, Huang, Ji, and Han (2019) proposed to employ Tree-LSTM to encode the dependency tree of the input text. Tree-LSTM is a version of LSTM with the key difference that at each time step, the hidden states of the Tree-LSTM neurons are updated using the representation of the current word and the hidden states of all of its children in the input tree structure. In addition to the dependency tree encoded by Tree-LSTM, authors also proposed to encode the external knowledge encoded in a domain-specific knowledge base (KB) using gating mechanism added to the Tree-LSTM update rules. More specifically, firstly, for each entity in the input text, their types and descriptions are obtained from the knowledge base. Next, the entity type and description are represented using randomly-initialized embedding of their words. Note that these embeddings will be fine-tuned during training. Afterwards, using the pooled representation of the entity type and description, a new gate vector is computed. The gate vector will be employed in the Tree-LSTM to control how much information should be transferred from the children to the parent node at each time step.

Although the Tree-LSTM or GCN architectures employed in the above mentioned works are effective to capture the structure of the input text, the

performance of these models will degrade by increasing the number of layers. This limitation prevent the model from encoding longer dependencies in the graph structure. To overcome this issue, authors in Yan, Jin, Meng, Guo, and Cheng (2019a) proposed to encode multi-order graph structure. More specifically, they compute the graph-based representation of the input text by employing the dependency tree adjacency matrix A, the second order of the adjacency matrix A^2 and the third-order of the adjacency matrix A^3. The aforementioned adjacency matrices will be encoded using graph attention network (GAT) which is a variant of GCN. The representation obtained for each order will be aggregated using attention function atop the proposed GATs.

Another issue with prior work is that they utilize dependency tree for event extraction while ignoring the dependency relation type between words. More specifically, the dependency tree is encoded using a binary adjacency matrix in which an entry is set to 1 if there is a dependency edge between the corresponding words, otherwise the entry is set to zero. To solve this limitation, in the proposed mode by Cui et al. (2020), authors suggest to model the structure of the sentence by encoding the dependency relations between words. More specifically, instead of using a binary adjacency matrix to encode the dependency tree, authors employ the tensor E of the dimension $n \times n \times p$ whose entry $E_{i,j}$ is a vector of size p, i.e., the total number of dependency relations in the dependency tree. Moreover, each edge in the dependency tree is represented with a randomly initialized vector. The words of the sentence are also encoded by the high-dimensional vectors obtained from a BiLSTM network. Next, to update the word representations, each channel of the adjacency tensor E is employed by a GCN layer to aggregate the representations of the neighbor nodes and connecting edges with the respect to the relation type

corresponding to the selected channel. Finally, using the updated representations of the nodes and the previous state of the edge vectors, the representation of each edge is updated using a feed forward neural net. By stacking of L layers of GCN and feed forward net to update the word representations and the edge representation, respectively, the final representation of the words is obtained. Finally, a feed forward classifier consumes the representations of the words to predict the event triggers.

1.6.4.2 Event Graph. A document might includes several event mentions. These events could have temporal, hierarchical or causal relations with each other. To construct the event graphs, prior works takes two major steps: (1) Event mention detection which identifies the events in the document, (2) Event-Event relation extraction which aims to predict the causal or temporal relations between events.

Recently, event-event relation extraction has gained more attention. For instance, authors in H. Wang, Chen, Zhang, and Roth (2020b) proposed joint model for simultaneously predicting the temporal and causal relations between event pairs using contextualized word embeddings and common sense knowledge injection. In particular, to pre-train a model for common sense knowledge injection, they propose to construct a set of positive and negative samples for event-event relations from two knowledge base ConceptNet and TemProb. Specifically, they extract 30,000 triples from these knowledge bases and annotate them using the relation specified between them in the knowledge base. They also construct another set of 30,000 triples in which there is no relation between the head and the tail based on the knowledge base facts. Afterwards, in a contrast learning framework, they train a multi-layer perception to distinguish the triples in which there is a

relation between them from the ones that are irrelevant to each other (i.e., with no relation between the head and the tail). Finally, during training of the event-event relation extraction model, the activations obtained from the pre-trained common sense knowledge network is employed as extra features to be concatenated with the features extracted for the input event-event paris. To obtain the event-event pair representations, the input document is first encoded by a pre-trained contextualized language model. Afterwards, the representations of the words in the document are concatenated with their POS tag embedding and are fed into a bi-directional LSTM (Bi-LSTM) model. Next, using the representation of the event triggers obtained from the Bi-LSTM layer and the features obtained by the pre-trained common sense knowledge network, the temporal and causal relations between every pair of events is predicted. The main advantage for joint temporal and causal relation extraction is that it could learn the features from one task that are indicative for the other task too.

In addition to temporal and causal relations between events, they might share their arguments and the arguments could have relations with each other too. These relations between events and their arguments could be encoded using graph structure. Identifying this graph could be helpful to recognize the co-occurring events and arguments for event extraction. Due to the importance of this task, recently it has gained attention M. Li, Zeng, et al. (2020); H. Wang, Chen, Zhang, and Roth (2020a) Specifically, authors in M. Li, Zeng, et al. (2020) proposed a language-model-based approach to construct the event graph between every pair of events from all documents in a corpus. In particular, they propose to find all possible connections between a pair of events using their mentions in multiple documents. Note that connection between two events refer to any path between the

events in the event graph that includes one or multiple arguments. After finding all possible connections, a language model (i.e., BERT model), pre-trained on the paths in the training set, predicts the importance of all found connections in test set. Finally, those connections that are above a threshold are selected to be used in the final event graph constructed for the two events of interest. Note that to predict the importance of a connection using the pre-trained language-model, authors propose to compute two types scores:

- Coherence and salience: This score evaluates the degree to which the candidate path is consistent with the two event types. For instance, the path *Attack, attacker, GPE, agent, Transport* should have high score with respect to coherence as it appears in the training data. To train the pre-train language-model to give high score to coherent paths, authors propose to train the BERT model in an autoregressive fashion (i.e., given the previous elements of a path, the model predicts the next element)

- Path co-occurrence: For a pair of events, some paths are more common and they frequently co-occur with each other. In order to train the language-model to give higher scores to the co-occurring paths, authors employ a contrasting learning objective. Specifically, they propose to construct an input sequence consisting of two paths for the BERT language model. If two paths belong to the same event graph, the input is labeled as positive, otherwise it is labeled as negative sample.

While the approach proposed by authors in M. Li, Zeng, et al. (2020) achieves promising results on construing the event graph, due to the breaking down of the event graph into paths, this model fail to capture any graph-level

interactions between edges and nodes in the event graph. As such, a potential direction for future work is to apply deep graph models to encode the event graph.

1.7 Disseration Outline

Given the importance of structure-based models for deep learning-based IE models, in this dissertation, we will study the novel methods of integrating structural information into deep learning models for IE. In particular, in Chapter 2, we discuss a novel mechanism to incorporate a diverse set of structural information for IE. Next, in Chapter 3, we study how task-specific structures can be inferred to improve IE performance. Moreover, we demonstrate novel methods to ensure consistency among all inferred structures. Afterward, in Chapter 4, we analyze structural information that is inferred beyond sample text and provide a view over all data points in a batch of data. Finally, the last Chapter provides a detailed case study in which structural information is helpful to overcome challenges in a less-explored IE setting, i.e., cross-lingual event extraction.

CHAPTER II

EXPLOITING DIVERSE STRUCTURES FOR INFORMATION EXTRACTION

This Chapter contains materials from the following published paper: "*Amir Pouran Ben Veyseh, Franck Dernoncourt, Quan Tran, Varun Manjunatha, Lidan Wang, Rajiv Jain, Doo Soon Kim, Walter Chang, and Thien Huu Nguyen.* '***Inducing rich interaction structures between words for document-level event argument extraction.***' *In Advances in Knowledge Discovery and Data Mining: 25th Pacific-Asia Conference, PAKDD 2021, Virtual Event, May 11–14, 2021, Proceedings, Part II, pp. 703-715. Cham: Springer International Publishing, 2021.*". In this publication, the experiments were entirely done by the author of this dissertation, Amir Pouran Ben Veyseh. The other co-authors provided feedback regarding the experiments and results. Amir wrote the entire paper and Dr. Thien Huu Nguyen provided editorial feedback for this paper.

In this chapter we study how the existing structures can be exploited to improve the performance of IE models. In particular, these structures are obtained from pre-trained general-purpose parsers. As discussed in the previous Chapter, prior works has shownd the effectiveness of the syntactic or semantic structures for IE. However, an effective method to combine various types of structures has not been studied. Concretely, a model to efficiently combine structure-based information for the downstream application is missing. In this Chapter, we study a novel method to combine syntactic, semantic, and external knowledge structures for the task of Event Argument Extraction (EAE) which is a sub-task of Event Extraction (EE).

Event Extraction (EE) is an important and challenging task in Information Exaction (IE) that aims to identify instances of events (i.e., change of states

of real-world entities) in text. To this end, two subtasks should be solved: (1) Event Detection (ED) to recognize event-triggering expressions (verbal predicates or nominalizations, called event triggers/mentions), and (2) Event Argument Extraction (EAE) to identify entity mentions that are involved in events (event participants and spatio-temporal attributes, collectively known as event arguments). This work focuses on EAE, a relatively less-explored task for EE (compared to ED). Technically speaking, our EAE task takes as inputs an event trigger and an argument candidate (entity mention), seeking to predict the role that the argument candidate plays in the event mention associated with the trigger. A well performing EAE system will benefit various downstream applications such as Knowledge Base Construction and Question Answering.

Most of the recent work on EAE employs deep learning models to achieve state-of-the-art performance X. Wang, Wang, et al. (2019). Unfortunately, these models are often restricted to sentence-level EAE where event triggers and arguments appear in the same sentence. In real world scenarios, arguments of an event might have been presented in sentences other than the sentence that hosts the event trigger in the input document. For instance, in the EE dataset of the DARPA AIDA program (phase 1)[1], 38% of arguments has been shown to be outside the sentences containing the corresponding triggers, i.e., in the document-level context Ebner, Xia, Culkin, Rawlins, and Van Durme (2020). As such, it is of paramount importance to develop models that can extract arguments of event mentions over the entire documents to provide a more complete view of information for events in documents.

[1] `https://tac.nist.gov/2019/SM-KBP/data.html`

A major challenge in document-level EAE involves long document context that hinders the ability of models to effectively identify important context words (among long word sequences) and link them to event triggers and arguments for role prediction. Recently, a promising approach to address this document context modeling issue has been explored for other related tasks in IE Nan et al. (2020); Sahu, Christopoulou, Miwa, and Ananiadou (2019); Thayaparan, Valentino, Schlegel, and Freitas (2019) where document structures (i.e., direct interactions between parts of documents) are employed to facilitate the connections and reasoning between important context words for a prediction problem.

Thus, one simple solution towards utilizing document structures for EAE is to exert one of the existing document-level models that has been designed for other related tasks. However, in this work, we show that such prior models have critical constraints that should be addressed to better serve EAE. As such, the existing document-level models have only exploited some (typically one) specific types of information/heuristics to form the edges in document structures, thus failing to leverage a diversity of useful information to enrich document structures in EAE. This is unfortunate as multiple information sources are often required simultaneously to capture necessary interaction information between nodes/words and improve the coverage/performance for EAE models. For instance, consider the following document: "*The foundation said that immediately following the Haitian earthquake, the Embassy of Algeria provided an unsolicited lump-sum fund to the foundation's relief plan. This was a one-time, specific donation to help Haiti and it had* ***donated*** *twice to the Clinton Foundation before.*". In this two-sentence document, an EAE system needs to recognize the entity mention "*Embassy of Algeria*" as an argument (of role *Giver*) for the event mention associated with

the trigger word "*donated*". To perform this reasoning, the models can utilize the coreference link between "*Embassy of Algeria*" and the pronoun "*it*" (i.e., discourse information) that can be directly connected with the trigger word "*donated*" via an edge in the syntactic dependency tree of the second sentence. Alternatively, if the coreference link cannot be obtained (e.g., due to errors in the coreference resolution systems), EAE models can rely on the close semantic similarity between "*donated*" and "*provided an unsolicited lump-sum fund*" that can be further linked to "*Embassy of Algeria*" via a dependency edge in the first sentence. As such, document-level models might need to jointly capture the information from syntax, semantic, and discourse structures to sufficiently encode necessary interactions between words for EAE.

Motivated by this intuition, we propose to combine different information sources to generate effective document structures for our EAE problem, focusing on the knowledge from syntax (i.e., dependency trees), discourse (i.e., coreference links), and semantic similarity. Importantly, for the semantic similarity, in addition to using contextualized representation vectors to compute interaction scores between words as in prior work Nan et al. (2020), we propose to further leverage external knowledge bases to enrich document structures for EAE. As such, we link the words in the documents to the entries in some external knowledge bases and exploit the entry similarity in such knowledge bases to obtain word similarity scores for the structures. To our knowledge, this is the first work to employ external knowledge bases to compute document structures for an IE task in the literature.

Given various document structures, how can we effectively combine these structures for EAE? Our main principle for this goal is motivated from the running example where the role reasoning process for the event trigger and

argument candidate involves a sequence of interactions with multiple other words, possibly using different types of information at each interaction step, e.g., syntax, discourse or semantic information (called heterogeneous interaction types). To this end, we propose to employ Graph Transformer Networks (GTN) Yun et al. (2019) to facilitate the implementation of this multi-hop heterogeneous reasoning idea. More specifically, GTNs fulfill the multi-hop heterogeneous reasoning by multiplying weighted sums of different initial document structures to generate rich combined structures. Finally, the resulting combined structures will be used to learn representation vectors for EAE based on graph convolutional networks (GCN). To our knowledge, this is also the first work that introduces GTN and GCN for document structure computation and representation learning in document-level EAE.

We evaluate the proposed model on two benchmark datasets; one for document-level EAE Ebner et al. (2020) and one for the closely related task of implicit semantic role labeling. Our experiments demonstrate the effectiveness of the proposed model, establishing new state-of-the-art results on both benchmark datasets.

2.1 Model

We formulate document-level EAE as a multi-class classification problem. The input to the models is a document $D = w_1, w_2, \ldots, w_N$ which consists of multiple sentences, i.e., S_i's. To be comparable with previous work Ebner et al. (2020), we also use a golden event trigger, i.e., the t-th word of D (w_t), and an argument candidate, i.e., the a-th word of D (w_a), as the inputs (w_t and w_a can occur in different sentences). The goal of EAE is to predict the role of the argument candidate w_a in the event evoked by w_t. Here, the role might be $None$,

indicating that w_a is not a participant in the event mention w_t. Our model for EAE involve three major components: (i) Document Encoder to transform the words in D into high dimensional vectors, (ii) Structure Generation to generate initial document structures for EAE, and (iii) Structure Combination to combine the structures and learn representation vectors for role prediction. We provide details for these components below.

2.1.1 Document Encoder. In the first step, we transform each word $w_i \in D$ into a representation vector x_i that is the concatenation of the following two vectors:

(i) The pre-trained word embedding of w_i: Here, we consider both non-contextualized embeddings, i.e., GloVe and contextualized embeddings, i.e., BERT in the experiments. In particular, for BERT, as w_i might be split into multiple word-pieces, we use the average of the hidden vectors for the word-pieces of w_i in the last layer as the word embedding vector for w_i. Following Ebner et al. (2020), we employ the BERT_{base} version that divides D into segments of 512 word-pieces to be encoded separately. In our experiments, we fix the parameters of the BERT_{base}.

(ii) The position embeddings of w_i: These vectors are obtained by looking up the relative distances between w_i and the trigger and argument words (i.e., $i - t$ and $i - a$ respectively) in a position embedding table. This table is initialized randomly and updated in the training process. Position embedding vectors are important as they notify the model about the positions of the trigger and argument words.

Given the vector sequence $X = x_1, x_2, \ldots, x_N$ to represent the words in D, we further send it to a bidirectional long short-term memory network (LSTM) to generate a more abstract vector sequence $H = h_1, h_2, \ldots, h_N$. Here, h_i is the

hidden vector for w_i that is obtained by concatenating the corresponding forward and backward hidden vectors from the bidirectional LSTM. We will use the hidden vectors in H as the inputs for the next computation. Note that we do not include the sentence boundary information of D into the hidden vectors H so far as it will be addressed in our document structures later.

2.1.2 Structure Generation. The goal of this section is to generate initial document structures that will be combined to learn representation vectors for document-level EAE in the next step. Formally, a document structure in our work involves an interaction graph $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$ between the words in D, i.e., $\mathcal{N} = \{w_i | w_i \in D\}$. As such, the document structure $\mathcal{G}$ can be represented via a real-valued adjacency matrix $A = \{a_{ij}\}_{i,j=1..N}$ where the value/score a_{ij} reflects the importance (or the level of interaction) of w_j for the representation computation of w_i for EAE. As presented in the introduction, we simultaneously consider three types of information to form the edges $\mathcal{E}$ (or compute the interaction scores a_{ij}) in this work, including syntax, semantics, and discourse. We describe initial document structures based on these information types in the following.

Syntax-based Structures: The motivation for this type of document structures is based on sentence-level EAE where dependency parsing trees of input sentences have been used to reveal important context, i.e., via shortest dependency paths to connect event triggers and arguments, and guide the interaction modeling between words for argument role prediction. As such, we expect dependency trees for sentences in D can also be exploited to provide useful information for document structures for EAE. In particular, we propose to leverage dependency relations/connections between pairs of words in D to compute the interaction scores a_{ij}^{dep} in the syntax-based document structure $A^{dep} = \{a_{ij}^{dep}\}_{i,j=1..N}$ for D. Here,

two words are more important to each other for representation learning if they are connected in dependency tress. To this end, we first obtain the dependency tree T_i for each sentence S_i in D using an off-the-shelf dependency parser[2]. Afterward, to connect the dependency trees T_i for the sentences, following Gupta, Rajaram, Schütze, and Runkler (2019), we create a link between the root node of a tree T_i for S_i with the root node of the tree T_{i+1} for the subsequent sentence S_{i+1}. The resulting graph with linked trees T_i is denoted by T^D. In the next step, motivated by shortest dependency paths in sentence-level EAE, we retrieve the shortest path P^D between the nodes for w_t and w_a in T^D. Finally, we compute the interaction score a_{ij}^{dep} by setting it to 1 if (w_i, w_j) or (w_j, w_i) is an edge in P^D, and 0 otherwise.

Semantic-based Structures: These document structures aim to evaluate the interaction scores in the structures based on the semantic similarity between words (i.e., two words are more important for the representation learning of each other if they are more semantically related). As such, we consider two complementary approaches to capture the semantics of the words in D for semantic-based structure generation, i.e., contextual semantics and knowledge-based semantics.

First, in contextual semantics, we seek to reveal the semantic of a word via the context in which it appears. This suggests the use of the contextualized representation vectors $h_i \in H$ (obtained from the LSTM model) to capture contextual semantics for the words $w_i \in D$ and produce the contextual semantic-based document structure $A^{context} = \{a_{ij}^{context}\}_{i,j=1..N}$ for D. Accordingly, to compute the semantic-based interaction score $a_{ij}^{context}$ for w_i and w_j, we employ

[2]We use the Stanford Core NLP Toolkit to parse the sentences in this work.

the normalized similarity score between their contextualized representation vectors:

$$k_i = U_k h_i, q_i = U_q h_i$$
$$a_{ij}^{context} = \exp(k_i q_j) / \sum_{v=1..N} \exp(k_i q_v) \tag{2.1}$$

where U_k and U_q are trainable weight matrices, and the biases are omitted in this work for brevity.

Second, in knowledge-based semantics, our goal is to employ the external knowledge of the words from knowledge bases to capture their semantics. We expect that such external knowledge can provide a complementary source of information for the contextual semantics of the words (i.e., external knowledge vs internal context), thereby enriching the document structures for D. To this end, we propose to utilize WordNet, a rich knowledge base for word meanings, to obtain external knowledge for the words in D. Essentially, WordNet involves a network that connects word meanings (i.e., synsets) according to various semantic relations (e.g., synonyms, hyponyms). Each node/synset in WordNet is associated with a textual glossary to provide expert definition about the corresponding meaning.

Our first step to generate knowledge-based document structures for D is to map each word $w_i \in D$ to a synset node M_i in WordNet that can be done with a Word Sense Disambiguation (WSD) tool. In this work, we use WordNet 3.0 and the state-of-the-art BERT-based WSD tool in Blevins and Zettlemoyer (2020) to perform such word-synset mapping. Afterward, to determine knowledge-based interaction scores between two words w_i and w_j in D, we can leverage the similarity scores between the two linked synset nodes M_i and M_j in WordNet. As such, to leverage the rich information embedded in the synset nodes M_i, we introduce two versions of knowledge-based document structures for D based on the glossaries of the synset nodes and the hierarchy structure in WordNet:

(1) The glossary-based structure $A^{gloss} = \{a_{ij}^{gloss}\}_{i,j=1..N}$: Here, for each word $w_i \in D$, we first retrieve the glossary GM_i from the corresponding linked node M_i in WordNet (GM_i can be seen as a sequence of words). A representation vector VM_i is then computed to capture the semantic information in GM_i, by applying the max-pooling operation over the pre-trained GloVe embeddings of the words in GM_i. Finally, the glossary-based interaction score a_{ij}^{gloss} for w_i and w_j is estimated via the similarity between the glossary representations VM_i and VM_j (with the consine similarity): $a_{ij}^{gloss} = cosine(VM_i, VM_j)$.

(2) The WordNet hierarchy-based structure $A^{struct} = \{a_{ij}^{struct}\}_{i,j=1..N}$: The interaction score a_{ij}^{struct} for w_i and w_j in this case relies on the structure-based similarity of the linked synset nodes M_i and M_j in WordNet. Accordingly, we employ the Lin similarity measure for the synset nodes in WordNet for this purpose: $a_{ij}^{struct} = \frac{2*\text{IC}(\text{LCS}(M_i,M_j))}{\text{IC}(M_i)+\text{IC}(M_j)}$ where IC and LCS represent the information content of the synset nodes and the least common subsumer of the two synsets in the WordNet hierarchy (most specific ancestor node), respectively.

Discourse-based Structures: Besides the typical lengths of the input texts, a key difference between document-level and sentence-level EAE involves the presence of multiple sentences in document-level EAE where discourse information (i.e., where the sentences span and how they relate to each other) is helpful to understand the input documents. The goal of this part is to leverage such discourse structures to provide complementary information for the syntax- and semantic-based document structures for EAE. To this end, we propose to exploit two following types of discourse information to generate discourse-based document structures for EAE: (1) the sentence boundary-based document structure $A^{sent} = \{a_{ij}^{sent}\}_{i,j=1..N}$: This document structure concerns the same sentence

information of the words in D. The intuition is that two words in the same sentence would involve more useful information for the representation computation of each other than those in different sentences. To implement this intuition, we compute A^{sent} by setting the sentence boundary-based score a_{ij}^{sent} to 1 if w_i and w_j appear in the same sentence in D and 0 otherwise; and (2) the coreference-based document structure $A^{coref} = \{a_{ij}^{coref}\}_{i,j=1..N}$: Instead of considering within-sentence information as in A^{sent}, this document structure exploits relations/connections between sentences (cross-sentence information) in D. To this end, we consider two sentences in D as being related if they contain entity mentions that refer to the same entity in D (coreference information)[3]. Given such a relation between sentences, we consider two words in D to be more relevant to each other if they appear in related sentences. To this end, for the coreference-based structure, a_{ij}^{coref} is set to 1 if w_i and w_j appear in different, but related sentences; and 0 otherwise.

2.1.3 Structure Combination. Up to this point, we have generated six different document structures for D (i.e., $\mathcal{A} = [A^{dep}, A^{context}, A^{gloss}, A^{struct}, A^{sent}, A^{coref}]$). As these document structures are based on complementary types of information (called structure types), this section aims to combine them to generate richer document structures for EAE. Our key intuition to achieve such a combination is to note that each importance score a_{ij}^{v} in one of the structures A_{ij}^{v} ($v \in V = \{dep, context, gloss, struct, sent, coref\}$) only considers the direct interaction between the two involving words w_i and w_j (i.e., not including any other words) according to one specific information type v. As motivated in the introduction, we expect each importance score in the combined structures to further condition on interactions with other important context words

[3]We use the Stanford Core NLP Toolkit to determine the coreference of entity mentions.

in D (i.e., in addition to the two involving words) where each interaction between a pair of words can flexibly use any of the six structure types (multi-hop and heterogeneous-type reasoning). To this end, we propose to use Graph Transformer Networks (GTN)Yun et al. (2019) to enable such a multi-hop and heterogeneous-type reasoning in the structure combination for EAE.

In particular, to enable multi-hop reasoning paths at different lengths, we first add the identity matrix I (of size $N \times N$) into the set of initial document structures $\mathcal{A} = [A^{dep}, A^{context}, A^{gloss}, A^{struct}, A^{sent}, A^{coref}, I] = [\mathcal{A}_1, \ldots, \mathcal{A}_7]$. The GTN model is then organized into C channels for structure combination, where the k-th channel contains M intermediate document structures $Q^k_1, Q^k_2, \ldots, Q^k_M$ of size $N \times N$. As such, each intermediate structure Q^k_i is computed by a linear combination of the initial structures in $\mathcal{A}$ using learnable weights α^k_{ij}: $Q^k_i = \sum_{j=1..7} \alpha^k_{ij} \mathcal{A}_j$. Here, due to the linear combination, the interaction scores in Q^k_i are able to reason with any of the six initial structure types in V (although such scores still consider the direct interactions of the involving words only). Afterward, the intermediate structures $Q^k_1, Q^k_2, \ldots, Q^k_M$ in each channel k are multiplied to generate the final document structure $Q^k = Q^k_1 \times Q^k_2 \times \ldots \times Q^k_M$ of size $N \times N$ (for the k-the channel). As shown in Yun et al. (2019), the matrix multiplication enables the importance score between a pair of words w_i and w_j in Q^k to condition on the multi-hop interactions/reasoning between the two words and other words in D (up to $M - 1$ hops due to the inclusion of I in $\mathcal{A}$). The interactions involved in one importance score in Q^k can also realize any of the initial structure types in V (heterogeneous reasoning) due to the flexibility of the intermediate structure Q^k_i.

Given the rich document structures $Q^1, Q^2, \ldots, Q^C$ from the C channels, GTN then feed them into C graph convolutional networks (GCN) Kipf and Welling

(2016) to induce document structure-enriched representation vectors for argument role prediction in EAE (one GCN for each $Q^k = \{Q^k_{ij}\}_{i,j=1..N}$). As such, each of these GCN models involve G layers that produces the hidden vectors $\bar{h}^{k,t}_1, \ldots, \bar{h}^{k,t}_N$ at the t-th layer of the k-th GCN model for the words in D ($1 \leq k \leq C$, $1 \leq t < G$):

$$\bar{h}^{k,t}_i = ReLU(U^{k,t} \sum_{j=1..N} \frac{Q^k_{ij} \bar{h}^{k,t-1}_j}{\sum_{u=1..N} Q^k_{iu}}) \tag{2.2}$$

where $U^{k,t}$ is the weight matrix for the t-th layer of the k-th GCN model and the input vectors $\bar{h}^{k,0}_i$ for the GCN models are obtained from the contextualized representation vectors H (i.e., $\bar{h}^{k,0}_i = h_i$ for all $1 \leq k \leq C$, $1 \leq i \leq N$).

In the next step, the hidden vectors in the last layers of all the GCN models (at the G-th layers) for w_i (i.e., $\bar{h}^{1,G}_i, \bar{h}^{2,G}_i, \ldots, \bar{h}^{C,G}_i$) are concatenated form the final representation vector h'_i for w_i in the proposed GTN model: $h'_i = [\bar{h}^{1,G}_i, \bar{h}^{2,G}_i, \ldots, \bar{h}^{C,G}_i]$.

Finally, to predict the argument role for w_a and w_t in D, we assemble a representation vector R based on the hidden vectors for w_a and w_t from the GTN model via: $R = [h'_a, h'_t, MaxPool(h'_1, h'_2, \ldots, h'_N)]$. This vector is then sent to a two-layer feed-forward network with softmax in the end to produce a probability distribution $P(.|D, a, t)$ over the possible argument roles. We then optimize the negative log-likelihood L_{pred} to train the model: $\mathcal{L} = -\log P(y|D, a, t)$ where y is the golden argument role for the input example. We call the proposed model the Multi-hop Reasoning for Event Argument extractor with heterogeneous Document structure types (MREAD) for convenience.

2.2 Experiments

Dataset & Parameters: We evaluate the document-level EAE models in this work on RAMS, a recent dataset introduced in Ebner et al. (2020) for

document-level EAE. RAMS contains 9,124 annotated event mentions across 139 types for 65 argument roles, serving as the largest available dataset for document-level EAE. We use the official train/dev/test split and evaluation script for RAMS provided by Ebner et al. (2020) to achieve a fair comparison. In addition, we evaluate the models on the BNB dataset Gerber and Chai (2012) for implicit semantic role labelling (iSRL), a closely related task to document-level EAE where the models need to predict roles of argument candidates for a given predicate (arguments and predicates can appear in different sentences in iSRL). In our experiments, we use the version of BNB prepared by Ebner et al. (2020) (with the same data split and pre-processing script) for a fair comparison. This dataset annotates 2,603 argument mentions for a total of 12 argument roles (for 1,247 predicates/triggers). We use the development set of the RAMS dataset to fine-tune the hyper-parameters of the proposed model MREAD.

Results: We compare our model MREAD with two categories of baselines on RAMS:

(1) Structure-free: These baselines do not exploit document structures for EAE. In particular, we compare MREAD with the RAMS_{model} model in Ebner et al. (2020) and the **Head-based** model in Z. Zhang, Kong, Liu, Ma, and Hovy (2020). Here, RAMS_{model} currently has the state-of-the-art (SOTA) performance for document-level EAE on RAMS.

(2) Structure-based: These baselines employ some forms of document structures (mostly based on syntax and semantic information) to learn representation vectors for input documents. Note that as none of the prior work has explored document structure-based models for document-level EAE, we compare MREAD with the existing document structure-based models for a related

Model	Standard Decoding			Type Constrained		
	P	R	F1	P	R	F1
RAMS	62.8	74.9	68.3	78.1	69.2	73.3
Head-based	71.5	66.2	68.8	81.1	66.2	73.0
iDepNN	65.8	68.0	66.9	77.1	67.7	72.1
EoG	71.0	71.7	71.4	82.4	69.2	75.2
GCNN	72.2	72.8	72.5	85.1	69.4	76.5
LSR	72.6	73.6	73.1	83.9	71.4	77.2
MREAD (ours)	75.7	75.3	**75.5**	88.2	72.1	**79.3**

Table 4. Performance on the RAMS test set using BERT.

task of document-level relation extraction (DRE) in IE. As such, the following SOTA models for DRE are considered in this category: (i) **iDepNN** Gupta et al. (2019); (ii) **GCNN** Sahu et al. (2019): This baseline generates document structures based on both syntax and discourse information (e.g., dependency trees, coreference links). Note that although GCNN also considers more than one source of information for document structures as we do, it fails to exploit semantic-based document structures (for both contextual and knowledge-based semantics) and lacks effective mechanisms for structure combination (i.e., not using GTN); (iii) **LSR** Nan et al. (2020); and (iv) **EoG** Christopoulou et al. (2019).

In addition to the standard decoding (i.e., using `argmax` with $P(.|D, a, t)$ to obtain the predicted roles), following Ebner et al. (2020), we also consider the decoding setting where the models' predictions are constrained to the permissible roles for the event type e evoked by the trigger w_t. Tables 4 and 5 show the the models' performance on the RAMS test set using BERT and GloVe embeddings, respectively. There are several observations from these tables. First, the proposed model MREAD significantly outperforms all the baselines in both the standard and type constrained decoding regardless of the used embeddings (BERT or GloVe). This consistent performance improvement is significant with $p < 0.01$ and clearly

demonstrates the effectiveness of MREAD for document-level EAE. Second, except for iDepNN, all the structure-based models significantly outperform the structure-free baselines. This finding is significant especially considering that the structure-based models are not originally designed for document-level EAE, thereby clearly showing the benefits of document structures for document-level EAE. Finally, compared to GCNN and EoG that also consider multiple sources of information as our model, MREAD achieves substantially better performance, suggesting the advantages of contextual and knowledge-based structures along with multi-hop heterogeneous reasoning in our EAE problem.

Model	Standard Decoding			Type Constrained		
	P	R	F1	P	R	F1
RAMS	66.3	69.8	68.0	77.4	68.8	72.9
Head-based	70.2	63.4	66.6	74.6	65.3	69.6
iDepNN	65.7	65.4	65.5	75.7	63.2	68.9
EoG	69.2	69.0	69.1	81.3	68.0	74.1
GCNN	71.1	70.9	71.0	83.7	68.1	75.1
LSR	72.5	72.0	72.2	82.9	70.3	76.1
MREAD (ours)	73.6	73.5	**73.5**	86.7	71.0	**78.1**

Table 5. Performance on the RAMS test set using GloVe.

Finally, we evaluate the performance of MREAD on the BNB dataset for iSRL. As we use the data version prepared by Ebner et al. (2020) that involves a different train/dev/test split from the original BNB dataset in Gerber and Chai (2012), we directly use the RAMS_{model} model in Ebner et al. (2020) as our baseline for a fair comparison. In addition, we report the performance of the structure-based baselines (iDepNN, GCNN, LSR, and EoG) for a complete view. Table 6 shows the performance of the models on the BNB test dataset (using BERT embeddings). As can be seen, MREAD is also better than all the baseline models substantially and

significantly ($p < 0.01$), further confirming the benefits of our proposed model in this work.

Model	P	R	F1
RAMS	-	-	76.6
iDepNN	80.0	75.1	77.5
EoG	78.5	74.4	76.4
GCNN	81.0	73.9	77.3
LSR	80.3	74.1	77.1
MREAD (ours)	82.9	75.0	**78.8**

Table 6. Performance on the BNB test set for iSRL.

Ablation Study: Our proposed model combines different types of document structures (i.e., six types in $\mathcal{A}$) using GTN to enable multi-hop and heterogeneous reasoning for document-level EAE. This section studies the contribution of the proposed document structures and structure combination in MREAD by evaluating the performance of the ablated versions of the model on the development set of the RAMS dataset. In particular, the following ablated models are examined: (i) **MREAD-**A^v: In this group of ablated models, we eliminate each of the document structures in $\mathcal{A}$ from MREAD and evaluate the performance of the model with the remaining structures (e.g., **MREAD-**A^{dep}, **MREAD-**A^{sent}, etc.), (ii) **MREAD-GTN**: In this ablated model, the GTN architecture is excluded from MREAD, so the GCN models are directly and separately applied to each document structure in $\mathcal{A}$. (iii) **MREAD-Multi-hop**: This ablated model is to show the effectiveness of multi-hop heterogeneous reasoning/interaction for EAE. As such, this model avoids the multiplication of the intermediate structures Q_i^k in each channel of GTN, and directly runs the GCN models over the intermediate document structures Q_i^k (i.e., the final structures Q^k are not produced).

Table 27 presents the performance of the models on the RAMS development set. As can be seen from the table, the removal of any document structures in

$\mathcal{A}$ would significantly hurt the performance of MREAD, thus confirming the effectiveness of the introduced document structures for EAE. Also, the significantly better performance of MREAD over MREAD-Multi-hop suggests that the multiplication of the intermediate structures in the channels of GTN is helpful to generate richer structures for EAE (i.e., by enabling multi-hop heterogeneous reasoning/interactions of words).

Model	P	R	F1
MREAD	75.5	76.5	**76.0**
MREAD-A^{def}	73.5	74.9	74.2
MREAD-$A^{context}$	72.7	73.5	73.1
MREAD-A^{gloss}	74.6	73.4	74.0
MREAD-A^{struct}	74.1	74.3	74.2
MREAD-A^{sent}	72.8	73.2	73.0
MREAD-A^{coref}	73.2	74.9	74.1
MREAD-GTN	72.1	73.7	72.9
MREAD-Multi-Hop	73.2	74.6	73.9

Table 7. Performance of the models on the RAMS development set using BERT embeddings and standard decoding.

2.3 Further Experiments on iSRL

In order to better assess the generalization ability of the proposed model to other datasets, we further evaluate the performance of MREAD on another iSRL dataset, i.e., SemEval 2010 Task 10 Ruppenhofer, Sporleder, Morante, Baker, and Palmer (2010). Similar to BNB, SemEval 2010 dataset annotates the predicates and their arguments across sentences (i.e., implicit arguments). This dataset contains 8,000 argument annotation for a total of 410 argument roles (for 1,819 predicates). We use the same data split and evaluation scripts provided by Ruppenhofer et al. (2010) for a fair comparison. As such, we compare the performance of MREAD with prior iSRL models that report their performance on the SemEval 2010 Task 10 dataset. The GloVe embeddings are used in this experiment to make it more compatible with prior work. Namely, we consider the

following baselines: **S&F** Silberer and Frank (2012), **VEC** and **Ensemble VEC** Gorinski, Ruppenhofer, and Sporleder (2013), **L&R** Laparra and Rigau (2013), **F&P** Feizabadi and Padó (2015), and the recent deep learning models **C&W Embedding** Schenk and Chiarcos (2016) and **NMSP** M. Le and Fokkens (2018). As can be seen in Table 8, MREAD can establish a new SOTA performance on the SemEval 2010 test set by improving the F1 score of the prior SOTA model by 1.4%.

Model	P	R	F1
VEC	21.0	18.0	19.0
VEC Ensemble	26.0	24.0	25.0
C&W Embedding	27.2	25.7	26.4
S&F	30.8	25.1	27.7
L&R	33.0	24.0	28.0
NMSP	28.8	28.6	28.7
F&P	35.0	30.8	32.8
MREAD (ours)	38.9	30.5	**34.2**

Table 8. Performance on the SemEval 2010 Task 10 test set for iSRL.

2.4 Error Analysis

In order to better understand the operation of the proposed model MREAD and provide suggestions for future improvement, this section analyzes the errors made by our model. As such, we sample 100 examples in the development set of RAMS that are incorrectly predicted by MREAD and manually categorize them into different types. For this analysis, we employ the best performing version of MREAD that is trained with BERT in the type constrained decoding setting. Seven categories of errors detected in our analysis along with their distribution are shown follows (Table 9 provides examples for each category and Figure 1 shows the error type distribution):

– **Related Roles** (38%): This type of errors involves examples where the golden roles are very similar to some other roles (e.g., *Target* vs. *Place*)

ID	Error	Sample	Prediction	Gold
1	Related Roles	We do not agree with what Russia is doing, **bombarding** Aleppo.	Place	Target
2	Lack of Entity Types	They're **importing** not only oil, but wheat and historic artefacts as well. Bilal Erdogan this week denied continuous Russian allegations of smuggling oil from ISIS.	Artifact	Transporter
3	Disconnected Sentences	3/12 Donald Trump's wife Melania delivered a speech at the GOP convention in Cleveland that was later found to have been cribbed in part from Michelle Obama's 2008 convention address AP. 5/12 Donald Trump held a joint press conference with Mexican leader Enrique Pena Nieto in Mexico City in August, hours before **reiterating** his harsh immigration plans at a campaign rally in Arizona Reuters	Communicator	Place
4	Background Knowledge	**Genocide** will never remain in the past. By recognizing the genocide, it will force the Turkish government to take a brave step and look into its own history," he said. Representatives from the Turkish and Armenian embassies were present in the German parliament while the vote was taking place.	Place	Victim
5	Imperfect Structures	Most synthetic drugs that end up in the United States come from China, either directly or by way of Mexico. Adding carfentanil to that list is likely to only diminish, not eliminate, global **supply**.	Destination	Origin
6	Infrequent Labels	In the wake of Snowden's disclosures about the NSA's snooping at home and abroad, the **spy** agency and other federal agencies sought to step up their controls on sensitive information.	Beneficiary	Spy
7	Incorrect Annotations	I would never have expected that he would have won the vast majority of people who **voted** in the Democratic primary under age 45.	Place	Ballot

Table 9. Examples for the types of errors. Trigger words are shown in **red bold** font while arguments are shown in blue underlined font.

and there is not enough evidence in the input documents to distinguish such similar roles. Among the similar roles for an example in this case, the model tends to choose the most frequent role (based on training data), thus causing an error when the most frequent role is different from the golden one. For instance, in the first example of Table 9 (ID 1), the correct role of the argument candidate "*Aleppo*" for the trigger word "*bombarding*" is

Target. However, due to the limitation of the context information, the model incorrectly predicts a similar and more frequent role of *Place* for the example.

– **Lack of Entity Types** (15%): This type of errors corresponds to the examples whose argument role predictions of the model are inconsistent with the entity types of the argument candidates (e.g., person, organization). Such errors could be avoided if the model exploits the entity type information of the argument candidates to capture the constraints between entity types and argument roles. For instance, in the second example of Table 9 (ID 2), the model incorrectly predicts the role *Artifact* for the argument candidate "*Bilal Erdogan*" of the event trigger "*importing*". This error could be fixed if the model realizes that "*Bilal Erdogan*" is a person and *Transporter* would be more suitable role for this entity type.

– **Disconnected Sentences** (14%): The examples in this type of errors involve sequences of captions for the photos in the input documents. Here, the syntax information and the discourse structures in these caption-based documents are different from those in the training data, which involves syntactically well-formed sentences with coherent structures, causing the failure of the model in this case. An example for this type of errors is shown in the third row of Table 9 (ID 3).

– **Background Knowledge** (20%): In this type of errors, the annotated role of an argument candidate for the given trigger word assumes some background knowledge that is not available in the document context. The model thus do not have enough information to make correct prediction in this case. For instance, in the fifth example of Table 9 (ID 4), the input

document does not provide any explicit evidence to infer that entity mention "*Armenian*" is an argument of the annotated role *Victim* for the event triggered by "*Genocide*". It thus suggests that some background knowledge, which is not available to the model, has been leveraged to perform this annotation, causing the failure of the model in this case.

– **Imperfect Structures** (8%): Our model is based on the document structures computed from syntax, semantic and discourse information to infer important context words and their interactions. Our model fails for the examples in this type of errors as the applied information cannot help the document structures to achieve their goal on capturing interactions of important context words. An example for this type of errors can be found in the sixth row of Table 9 (ID 5) with the argument role prediction for "*China*" in the event triggered by "*supply*". Here, both the syntax and discourse information cannot help to effectively link "*China*" and "*supply*" as the shortest dependency path between them is long and does not contain effective context words; and there is no coreference links between the corresponding sentences. In addition, the importance scores between "*China*" and "*supply*" as well as their neighboring words are also weak according to the semantic-based document structures (i.e., for both contextual and knowledge-based information), eventually hindering the model to make a correct prediction in this case.

– **Infrequent Labels** (3%): Some event types and their roles are very rare in the RAMS dataset (e.g., the *Spy* event). As such, it is likely that these rare event types and roles are dominated by others, causing the errors of the model in this case (see the row with ID 6 in Table 9 for an example).

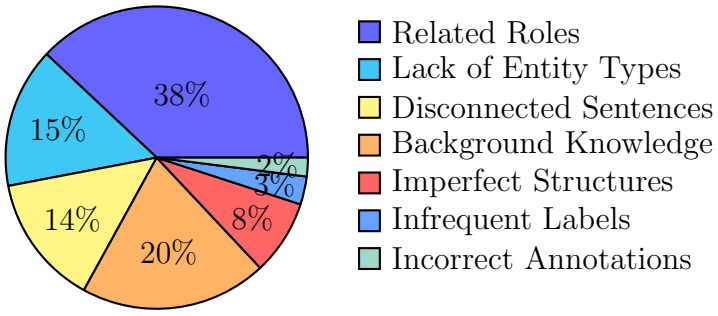

Figure 1. Distribution of error types in MREAD on the RAMS development set.

– **Incorrect Annotations** (2%): We also find that a small portion of the RAMS dataset is erroneously labeled, so the golden labels are incorrect while the predicted labels are actually correct in these samples. The last row in Table 9 (ID 7) shows an example of such erroneous annotations.

2.5 Related Work

Most of prior work on EE has focused on sentence-level EAE Lai, Nguyen, and Nguyen (2020a); D. M. Le and Nguyen (2021); Q. Li et al. (2013); T. H. Nguyen, Cho, and Grishman (2016); T. M. Nguyen and Nguyen (2019b); Pouran Ben Veyseh, Nguyen, and Nguyen (2020). Recently, some work has considered document-level EAE, featuring Ebner et al. (2020) as the most related work to our problem. However, the model proposed by Ebner et al. (2020) (i.e., RAMS_{model}) does not consider document structures to improve the performance for document-level EAE as we do in this work. Our work is also related to the recent document structure-based models for other NLP tasks Christopoulou et al. (2019); Thayaparan et al. (2019); H. M. Tran, Nguyen, and Nguyen (2020). However, compared to our proposed model, these prior works on document structures fail to exploit external knowledge to generate the structures and do not involve mechanisms to combine multiple structures for multi-hop heterogeneous reasoning.

2.6 Conclusion

This Chapter presents a novel deep learning model for document-level EAE. To facilitate the interaction of important context words in the documents for EAE, our model leverages multiple sources of information, including the novel employment of external knowledge bases, to generate document structures to provide effective knowledge for representation learning in EAE. Also, for the first time in EAE, graph transformer networks are employed to produce richer document structures. The experiments confirm the benefits of the proposed model, yielding to SOTA performance on benchamrk datasets.

While the proposed method is en effective solution for integrating different types of structural information for IE, it is limited to the structures that are obtained from existing parsers. In the next Chapter, we will study a novel method to infer structures that are not limited to the existence of pre-trained parsers. Concretely, the structural views are inferred based on the downstream IE task. Also, a novel mechanism is introduced to ensure consistency between all structures that are inferred.

CHAPTER III

INFERRING STRUCTURE FOR IE AT EXAMPLE LEVEL

This Chapter contains materials from the published papers "*Amir Pouran Ben Veyseh, Franck Dernoncourt, My Thai, Dejing Dou, and Thien Huu Nguyen. '**Multi-view consistency for relation extraction via mutual information and structure prediction.**' In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 05, pp. 9106-9113. 2020*" and "*Amir Pouran Ben Veyseh, Franck Dernoncourt, Dejing Dou, and Thien Huu Nguyen. '**Exploiting the syntax-model consistency for neural relation extraction.**' In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 8021-8032. 2020*". In these publications, the experiments were entirely done by the author of this dissertation, Amir Pouran Ben Veyseh. The other co-authors provided feedback regarding the experiments and results. Amir wrote the entire paper and Dr. Thien Huu Nguyen provided editorial feedback for this paper.

In the previous chapter, we discussed an effective method to combine multiple types of structural information for the task of document-level event argument extraction. One of the limitations of this approach is that it is limited to the existence of efficient parsers to obtain structural information for the task in hand. On the other hand, these structures are general-purpose and cannot provide effective interactions for the downstream IE task. In order to address these shortcomings, in this chapter, we study how the structural information can be inferred by the IE model, so that the important task-specific interactions are available for the IE model. In particular, we study the task of Relation Extraction. To this end, two methods that infer different structures for RE and ensure consistency between them are presented in this chapter. In particular,

we first introduce a method for inferring the structural information using novel architecture Ordered Neuron LSTM, then we provide an effective method based on Mutual Information to ensure consistency of the structural information that is inferred by the RE model.

3.1 Exploiting the Syntax-Model Consistency for Neural Relation Extraction

One of the fundamental tasks in Information Extraction (IE) is Relation Extraction (RE) where the goal is to find the semantic relationships between two entity mentions in text. Due to its importance, RE has been studied extensively in the literature. The recent studies on RE has focused on deep learning to develop methods to automatically induce sentence representations from data T. H. Nguyen and Grishman (2015a); Verga, Strubell, and McCallum (2018); D. Zeng et al. (2014). A notable insight in these recent studies is that the syntactic trees of the input sentences (i.e., the dependency trees) can provide effective information for the deep learning models, leading to the state-of-the-art performance for RE recently Guo, Zhang, and Lu (2019); V.-H. Tran, Phi, Shindo, and Matsumoto (2019); Xu et al. (2015b). In particular, the previous deep learning models for RE has mostly exploited the syntactic trees to structure the network architectures according to the word connections presented in the trees (e.g., performing Graph Convolutional Neural Networks (GCN) over the dependency trees Y. Zhang et al. (2018)). Unfortunately, these models might not be able to generalize well as the tree structures of the training data might significantly differ from those in the test data (i.e., the models are overfit to the syntactic structures in the training data). For instance, in the cross-domain setting for RE, the domains for the training data and test data are dissimilar, often leading to a mismatch between the syntactic

structures of the training data and test data. In order to overcome this issue, the overall strategy is to obtain a more general representation of the syntactic trees that can be used to inject the syntactic information into the deep learning models to achieve better generalization for RE.

A general tree representation for RE is presented in Veyseh et al. (2019) where the dependency trees are broken down into their sets of dependency relations (i.e., the edges) between the words in the sentences (called the edge-based representation). These dependency relations are then used in a multi-task learning framework for RE that simultaneously predicts both the relation between the two entity mentions and the dependency connections between the pairs of words in the input sentences. Although the dependency connections might be less specific to the training data than the whole tree structures, the major limitation of the edge-based representation is that it only captures the pairwise (local) connections between the words and completely ignores the overall (global) importance of the words in the sentences for the RE problem. In particular, some words in a given sentence might involve more useful information for relation prediction in RE than the other words, and the dependency tree for this sentence can help to better identify those important words and assign higher importance scores for them (e.g., choosing the words along the shortest dependency paths between the two entity mentions). We expect that introducing such importance information for the words in the deep learning models might lead to improved performance for RE. Consequently, in this work, we propose to obtain an importance score for each word in the sentences from the dependency trees (called the syntax-based importance scores). These will serve as the general tree representation to incorporate the syntactic information into the deep learning models for RE.

How can we employ the syntax-based importance scores in the deep learning models for RE? In this work, we first use the representation vectors for the words from the deep learning models to compute another importance score for each word (called the model-based importance scores). These model-based importance scores are expected to quantify the semantic information that a word contributes to successfully predict the relationship between the input entity mentions. Afterward, we propose to inject the syntax-based importance scores into the deep learning models for RE by enforcing that the model-based importance scores are consistent with the syntactic counterparts (i.e., via the KL divergence). The motivation of the consistency enforcement is to promote the importance scores as the bridge through which the syntactic information can be transmitted to enrich the representation vectors in the deep learning models for RE.

In order to implement this idea, we employ the Ordered-Neuron Long Short-Term Memory Networks (ON-LSTM) Shen, Tan, Sordoni, and Courville (2019a) to compute the model-based importance scores for the words in the sentences for RE. ON-LSTM extends the popular Long Short-Term Memory Networks (LSTM) by introducing two additional gates (i.e., the master forget and input gates) in the hidden vector computation. These new gates controls how long each neuron in the hidden vectors should be activated across different time steps (words) in the sentence (i.e., higher-order neurons would be maintained for a longer time). Based on such controlled neurons, the model-based importance score for a word can be determined by the number of active neurons that the word possesses in the operation of ON-LSTM. To our knowledge, this is the first time ON-LSTM is applied for RE in the literature.

One of the issues in the original ON-LSTM is that the master gates and the model-based importance score for each word are only conditioned on the word itself and the left context encoded in the previous hidden state. However, in order to infer the importance for a word in the overall sentence effectively, it is crucial to have a view over the entire sentence (i.e., including the context words on the right). To this end, instead of relying only on the current word, we propose to obtain an overall representation of the sentence that is used as the input to compute the master gates and the importance score for each word in the sentence. This would enrich the model-based importance scores with the context from the entire input sentences, potentially leading to the improved RE performance of the model in this work.

Finally, to further improve the representations learned by the deep learning models for RE, we introduce a new inductive bias to promote the similarity between the representation vectors for the overall sentences and the words along the shortest dependency paths between the two entity mentions. The intuition is that the relation between the two entity mentions of interest in a sentence for RE can be inferred from either the entire sentence or the shortest dependency path between the two entity mentions (due to the demonstrated ability of the shortest dependency path to capture the important context words for RE in the prior work Bunescu and Mooney (2005)). We thus expect that the representation vectors for the sentence and the dependency path should be similar (as both capture the semantic relation) and explicitly exploiting such similarity can help the models to induce more effective representations for RE. Our extensive experiments on three benchmark datasets (i.e., ACE 2005, SPOUSE and SciERC) demonstrate

the effectiveness of the proposed model for RE, leading to the state-of-the-art performance for these datasets.

3.1.1 Related Work. RE has been traditionally solved by the feature-based or kernel-based approaches Bunescu and Mooney (2005); Chan and Roth (2010); T. H. Nguyen and Grishman (2014); T. H. Nguyen, Plank, and Grishman (2015c); Sun et al. (2011); Zelenko et al. (2003); G. Zhou et al. (2005b). One of the issues in these approaches is the requirement for extensive feature or kernel engineering effort that hinder the generalization and applicability of the RE models. Recently, deep learning has been applied to address these problems for the traditional RE approaches, producing the state-of-the-art performance for RE. The typical network architectures for RE include the Convolutional Neural Networks dos Santos, Xiang, and Zhou (2015); T. H. Nguyen and Grishman (2015a); L. Wang et al. (2016); D. Zeng et al. (2014), Recurrent Neural Networks L. T. Nguyen, Van Ngo, Than, and Nguyen (2019); T. H. Nguyen and Grishman (2016); Y. Zhang et al. (2017); P. Zhou et al. (2016), and self-attentions in Transformer Verga et al. (2018). The syntactic information from the dependency trees has also been shown to be useful for the deep learning models for RE Guo et al. (2019); Y. Liu et al. (2015); Miwa and Bansal (2016); N. Peng et al. (2017); Y. Song, Wang, Jiang, Liu, and Rao (2019); Tai, Socher, and Manning (2015); V.-H. Tran et al. (2019); Veyseh et al. (2019); Xu et al. (2015b); Y. Zhang et al. (2018). However, these methods tend to poorly generalize to new syntactic structures due to the direct reliance on the syntactic trees (e.g., in different domains) or fail to exploit the syntax-based importance of the words for RE due to the sole focus on edges of the dependency trees Veyseh et al. (2019).

3.1.2 Model. The RE problem can be formulated as a multi-class classification problem. Formally, given an input sentence $W = w_1, w_2, \ldots, w_N$ where w_t is the t-th word in the sentence W of length N, and two entity mentions of interest at indexes s and o ($1 \leq s < o \leq N$), our goal is to predict the semantic relation between w_s and w_o in W.

Similar to the previous work on deep learning for RE Shi et al. (2018); Veyseh et al. (2019), we first transform each word w_t into a representation vector x_t using the concatenation of the three following vectors: (i) the pre-trained word embeddings of w_t, (ii) the position embedding vectors (to encode the relative distances of w_t to the two entity mentions of interest w_s and w_o (i.e., $t - s$ and $t - o$)), and (iii) the entity type embeddings (i.e., the embeddings of the BIO labels for the words to capture the entity mentions present in X). This word-to-vector transformation converts the input sentence W into a sequence of representation vectors $X = x_1, x_2, \ldots, x_N$ to be consumed by the next neural computations of the proposed model.

There are three major components in the RE model in this work, namely (1) the CEON-LSTM component (i.e., context-enriched ON-LSTM) to compute the model-based importance scores of the words w_t, (2) the syntax-model consistency component to enforce the similarity between the syntax-based and model-based importance scores, and (3) the similarity component between the representation vectors of the overall sentence and the shortest dependency path.

3.1.2.1 CEON-LSTM. The goal of this component is to obtain a score for each word w_t that indicates the contextual importance of w_t with respect to the relation prediction between w_s and w_o in W. In this section, we first describe the ON-LSTM model to achieve these importance scores (i.e., the model-based

scores). A new model (called CEON-LSTM) that integrates the representation of the entire sentence into the cells of ON-LSTM will be presented afterward.

ON-LSTM: Long-short Term Memory Networks (LSTM) Hochreiter and Schmidhuber (1997) has been widely used in Natural Language Processing (NLP) due to its natural mechanism to obtain the abstract representations for a sequence of input vectors M. Nguyen and Nguyen (2018b); T. M. Nguyen and Nguyen (2019a). Given the input representation vector sequence $X = x_1, x_2, \ldots, x_N$, LSTM produces a sequence of hidden vectors $H = h_1, h_2, \ldots, h_N$ using the following recurrent functions at the time step (word) w_t (assuming the zero vector for h_0):

$$\begin{aligned} f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f) \\ i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \\ o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o) \\ \hat{c}_t &= tanh(W_c x_t + U_c h_{t-1} + b_c) \\ c_t &= f_t \circ c_{t-1} + i_t \circ \hat{c}_t \\ h_t &= o_t \circ tanh(c_t) \end{aligned} \tag{3.1}$$

where f_t, i_t and o_t are called the forget, input and output gates (respectively).

In order to compute the importance score for each word w_t, ON-LSTM introduce into the mechanism of LSTM two additional gates, i.e., the master forget gate $\hat{f}_t$ and the master input gate $\hat{i}_t$ Shen et al. (2019a). These gates are computed

and integrated into the LSTM cell as follow:

$$
\begin{aligned}
\hat{f}_t &= cummax(W_{\hat{f}} x_t + U_{\hat{f}} h_{t-1} + b_{\hat{f}}) \\
\hat{i}_t &= 1 - cummax(W_{\hat{i}} x_t + U_{\hat{i}} h_{t-1} + b_{\hat{i}}) \\
\bar{f}_t &= \hat{f}_t \circ (f_t \hat{i}_t + 1 - \hat{i}_t) \\
\bar{i}_t &= \hat{i}_t \circ (i_t \hat{f}_t + 1 - \hat{f}_t) \\
c_t &= \bar{f}_t \circ c_{t-1} + \bar{i}_t \circ \hat{c}_t
\end{aligned}
\tag{3.2}
$$

where $cummax$ is an activation function defined as $cummax(x) = cumsum(softmax(x))$[1].

The forget and input gates in LSTM (i.e., f_t and i_t) are different from the master forget and input gates in ON-LSTM (i.e., $\hat{f}_t$ and $\hat{i}_t$) as the gates in LSTM assume that the neurons/dimensions in their hidden vectors are equally important and that these neurons are active at every step (word) in the sentence. This is in contrast to the master gates in ON-LSTM that impose a hierarchy over the neurons in the hidden vectors and limit the activity of the neurons to only a portion of the words in the sentence (i.e., higher-ranking neurons would be active for more words in the sentence). Such hierarchy and activity limitation are achieved via the function $cumax(x)$ that aggregates the softmax output of the input vector x along the dimensions. The output of $cumax(x)$ can be seen as the expectation of some binary vector of the form $(0, \ldots, 0, 1, \ldots, 1)$ (i.e., involving two consecutive segments: the 0's segment and the 1's segment). At one step, the 1's segments in the gate vectors represents the neurons that are activated at that step. In ON-LSTM, a word w_i is more contextually important than another word w_j if the master gates for w_i have more active neurons than those for w_j. Consequently, in order to compute the importance score for the word w_t, we can rely on the

[1] $cumsum(u_1, u_2, \ldots, u_n) = (u'_1, u'_2, \ldots, u'_n)$ where $u'_i = \sum_{j=1..i} u_j$.

number of active neurons in the master gates that can be estimated by the sum of the weights of the neurons in the master gates in ON-LSTM. Following Shen et al. (2019a), we employ the hidden vectors for the master forget gate in ON-LSTM to compute the importance scores for the words in this work. Specifically, let $\hat{f}_t = \hat{f}_{t1}, \hat{f}_{t2}, \ldots, \hat{f}_{tD}$ be the weights for the neurons/dimensions in $\hat{h}_t$ (i.e., D is the dimension of the gate vectors). The model-based importance score mod_t for the word $w_t \in W$ is then obtained by: $mod_t = 1 - \sum_{i=1..D} \hat{f}_{ti}$. For convenience, we also use $H = h_1, h_2, \ldots, h_N$ to denote the hidden vectors returned from the application of ON-LSTM over the input representation vectors X.

Introducing Sentence Context into ON-LSTM. One limitation of the ON-LSTM model is that it only relies on the representation vector of the current word x_t and the hidden vector for the left context (encoded in h_{t-1}) to compute the master gate vectors and the model-based important score for the word w_t as well. However, this score computation mechanism might not be sufficient for RE as the importance score for w_t might also depend on the context information on the right (e.g., the appearance of some word on the right might make w_t less important for the relation prediction between w_s and w_o). Consequently, in this work, we propose to first obtain a representation vector $x'_t = g(x_1, x_2, \ldots, x_N)$ that has the context information about the entire sentence W (i.e., both the left and right context for the current word w_t). Afterward, x'_t will replace the input representation vector x_t in the computation for the master gates and importance score at step t of ON-LSTM (i.e., in the formulas for $\hat{f}_t$ and $\hat{i}_t$ in Equation 3.2). In this way, the model-based importance score for w_t will be able to condition on the overall context in the input sentence.

In this work, we obtain the representation vector x'_t for each step t of ON-LSTM based on the weighted sum of the transformed vectors of the input representation sequence $x_1, x_2, \ldots, x_N$: $x'_t = \sum_i \alpha_{ti}(W_x x_i + b_x)$. The weight α_{ti} for the term with x_i in this formula is computed by:

$$\alpha_{ti} = \frac{\exp((W_h h_{t-1} + b_h) \cdot (W_x x_i + b_x))}{\sum_{j=1}^{N} \exp((W_h h_{t-1} + b_h) \cdot (W_x x_j + b_x))} \tag{3.3}$$

where W_h, b_h, W_x and b_x are the learnable parameters. Note that in this formula, we use the ON-LSTM hidden vector h_{t-1} from the previous step as the query vector to compute the attention weight for each word. The rationale is to enrich the attention weights for the current step with the context information from the previous steps (i.e., encoded in h_{t-1}), leading to the contextualized input representation x'_t with richer information for the master gates and importance score computations in ON-LSTM. The proposed ON-LSTM with the enriched input vectors x'_t is called CEON-LSTM (i.e., Context-Enriched ON-LSTM) in this work.

3.1.2.2 Syntax-Model Consistency.

As mentioned in the introduction, the role of the model-based importance scores obtained from CEON-LSTM is to serve as the bridge to inject the information from the syntactic structures of W into the representation vectors of the deep learning models for RE. In particular, we first leverage the dependency tree of W to obtain another importance score syn_t for each word $w_t \in W$ (i.e., the syntax-based importance score). Similar to the model-based scores, the syntax-based scores are expected to measure the contextual importance of w_t with respect to the relation prediction for w_s and w_o. Afterward, we introduce a constraint to encourage the consistency between the model-based and syntax-based importance scores (i.e., mod_t and syn_t) for the words via minimizing the KL divergence L_{import} between the normalized

scores:

$$\overline{mod}_1, \ldots, \overline{mod}_N = softmax(mod_1, \ldots, mod_N)$$
$$\overline{syn}_1, \ldots, \overline{syn}_N = softmax(syn_1, \ldots, syn_N) \quad (3.4)$$
$$\mathcal{L}_{import} = -\Sigma_i \overline{mod}_i log \frac{\overline{mod}_i}{\overline{syn}_i}$$

The intuition is to exploit the consistency to supervise the model-based importance scores from the models with the syntax-based importance scores from the dependency trees. As the model-based importance scores are computed from the master gates with the active and inactive neurons in CEON-LSTM, this supervision allows the syntactic information to interfere directly with the internal computation/structure of the cells in CEON-LSTM, potentially generating representation vectors with better syntax-aware information for RE.

To obtain the syntax-based importance scores, we take the motivation from the previous work on RE where the shortest dependency paths between the two entity mentions of interest have been shown to capture many important context words for RE. Specifically, for the sentence W, we first retrieve the shortest dependency path DP between the two entity mentions w_s and w_o and the length T of the longest path between any pairs of words in the dependency tree of W. The syntax-based importance score syn_t for the word $w_t \in W$ is then computed as the difference between T and the length of the shortest path between w_t and some word in DP in the dependency tree (i.e., the words along DP will have the score of T). On the one hand, these syntax-based importance scores are able to capture the importance of the words that is customized for the relation prediction between w_s and w_o. This is better suited for RE than the direct use of the edges in the dependency trees in Veyseh et al. (2019) that is agnostic to the entity mentions of interest and fails to encode the importance of the words for RE. On

the other hand, the syntax-based importance scores syn_t represent a relaxed form of the original dependency tree that might have a better chance to generalize over different data and domains for RE than the prior work (i.e., the ones that directly fit the models to the whole syntactic structures Y. Zhang et al. (2018) and run the risk of overfitting to the structures in the training data).

3.1.2.3 Sentence-Dependency Path Similarity.

In this component, we seek to further improve the representation vectors in the proposed deep learning model for RE by introducing a novel constraint to maximize the similarity between the representation vectors for the overall input sentence W and the words along the shortest dependency path DP (i.e., inductive bias). The rationale for this bias is presented in the introduction.

In order to implement this idea, we first obtain the representation vectors R_W and R_{DP} for the sentence W and the words along DP (respectively) by applying the max-pooling operation over the CEON-LSTM hidden vectors $h_1, h_2, \ldots, h_N$ for the words in W and DP: $R_W = max\_pooling_{w_i \in W}\{h_i\}$ and $R_{DP} = max\_pooling_{w_i \in DP}\{h_i\}$. In the next step, we promote the similarity between R_W and R_{DP} by explicitly minimizing their negative cosine similarity[2], i.e., adding the following term L_{path} into the overall loss function:

$$L_{path} = 1 - \cos\left(R_W, R_{DP}\right) \tag{3.5}$$

3.1.2.4 Prediction.

Finally, in the prediction step, following the prior work Veyseh et al. (2019), we employ the following vector V as the overall representation vector to predict the relation between w_s and w_o in W: $V = [x_s, x_o, h_s, h_o, R_W]$. Note that V involves the information at different abstract

[2]We tried the KL divergence and the mean square error for this, but cosine similarity achieved better performance.

levels for W, i.e., the raw input level with x_s and x_o, the abstract representation level with h_s and h_o from CEON-LSTM, and the overall sentence vector R_W. In our model, V would be fed into a feed-forward neural network with the softmax layer in the end to estimate the probability distribution $P(.|W, w_s, w_o)$ over the possible relations for W. The negative log-likelihood function is then obtained to serve as the loss function for the model: $L_{label} = -\log P(y|W, w_s, w_o)$ (y is the golden relation label for w_s and w_o in W). Eventually, the overall loss function of the model in this work is:

$$\mathcal{L} = \mathcal{L}_{label} + \alpha\mathcal{L}_{import} + \beta\mathcal{L}_{path} \tag{3.6}$$

where α and β are trade-off parameters. The model is trained with shuffled mini-batching.

3.1.3 Experiments.

3.1.3.1 Datasets and Hyper-parameters.

We evaluate the models in this work using three benchmark datasets, i.e., ACE 2005, SPOUSE, and SciERC. For ACE 2005, similar to the previous work L. Fu, Nguyen, Min, and Grishman (2017); T. H. Nguyen and Grishman (2016); Shi et al. (2018); Veyseh et al. (2019), we use the dataset preprocessed and provided by M. Yu et al. (2015) for compatible comparison. There are 6 different domains in this dataset, i.e., (`bc`, `bn`, `cts`, `nw`, `un`, and `wl`), covering text from news, conversations and web blogs. Following the prior work, the union of the domains `bn` and `nw` (called `news`) is used as the training data (called the source domain); a half of the documents in `bc` is reserved for the development data, and the remainder (`cts`, `wl` and the other half of `bc`) serve as the test data (called the target domains). This data separation facilitates the evaluation of the cross-domain generalization of the models due to the domain difference of the training and test data.

The SPOUSE dataset is recently introduced by Hancock et al. (2018), involving 22,195 sentences for the training data, 2,796 sentences for the validation data, and 2,697 sentences for the test data. Each sentence in this dataset contains two marked person names (i.e., the entity mentions) and the goal is to identify whether the two people mentioned in the sentence are spouses.

Finally, the SciERC dataset Luan, He, Ostendorf, and Hajishirzi (2018) annotates 500 scientific abstracts for the entity mentions along with the coreferences and relations between them. For RE, this dataset provides 3,219 sentences in the training data, 455 sentences in the validation data and 974 sentences in the test data.

We fine tune the hyper-parameters for the models in this work on the validation data of the ACE 2005 dataset. The best parameters suggested by this process include: 30 dimensions for the position embeddings and entity type embeddings, 200 hidden units for the CEON-LSTM model and all the other hidden vectors in the model (i.e., the hidden vectors in the final feed-forward neural network (with 2 layers) and the intermediate vectors in the weighted sum vector for x'_t), 1.0 for both loss trade-off parameters α and β, and 0.001 for the initial learning rate with the Adam optimizer. The batch size is set to 50. Finally, we use either the uncontextualized word embeddings `word2vec` (with 300 dimensions) or the hidden vectors in the last layer of the BERT_{base} model (with 768 dimensions) Devlin, Chang, Lee, and Toutanova (2019a) to obtain the pre-trained word embeddings for the sentences Devlin et al. (2019a). We find it better to fix BERT in the experiments.

3.1.3.2 Comparison with the state of the art. We fist compare the proposed model (called CEON-LSTM) with the baselines on the popular ACE

2005 dataset. In particular, the four following groups of RE models in the prior work on RE with the ACE 2005 dataset is chosen for comparison:

(i) Feature based models: These models hand-design linguistic features for RE, i.e., FCM, Hybrid FCM, LRFCM, and SVM Hendrickx et al. (2010); M. Yu et al. (2015).

(ii) Deep sequence-based models: These models employ deep learning architectures based on the sequential order of the words in the sentences for RE, i.e., log-linear, CNN, Bi-GRU, Forward GRU, Backward GRU T. H. Nguyen and Grishman (2016), and CNN+DANN L. Fu et al. (2017).

(iii) Adversarial learning model: This model, called GSN, attempts to learn the domain-independent features for RE Shi et al. (2018).

(iv) Deep structure-based models: These models use dependency trees either as the input features or the graphs to structure the network architectures in the deep learning models. The state-of-the-art models of this type include: AGGCN (Attention Guided GCN) Guo et al. (2019), SACNN (Segment-level Attention-based CNN) V.-H. Tran et al. (2019) and DRPC (the Dependency Relation Prediction and Control model) Veyseh et al. (2019). DRPC has the best reported performance on ACE 2005. Note that we obtain the performance of these models on the considered datasets using the actual implementation released by the original papers.

Most of the prior RE work on the ACE 2005 dataset uses the uncontextualized word embeddings (i.e., `word2vec`) for the initial word representation vectors. In order to achieve a fair comparison with the baselines, we first show the performance of the models (i.e., the F1 scores) on the ACE 2005 test datasets when `word2vec` is employed for the pre-trained word embeddings in

System	bc	cts	wl	Avg.
FCM M. Yu et al. (2015)	61.90	52.93	50.36	55.06
Hybrid FCM M. Yu et al. (2015)	63.48	56.12	55.17	58.25
LRFCM M. Yu et al. (2015)	59.40	-	-	-
Log-linear T. H. Nguyen and Grishman (2016)	57.83	53.14	53.06	54.67
CNN T. H. Nguyen and Grishman (2016)	63.26	55.63	53.91	57.60
Bi-GRU T. H. Nguyen and Grishman (2016)	63.07	56.47	53.65	57.73
Forward GRU T. H. Nguyen and Grishman (2016)	61.44	54.93	55.10	57.15
Backward GRU T. H. Nguyen and Grishman (2016)	60.82	56.03	51.78	56.21
CNN+DANN L. Fu et al. (2017)	65.16	-	-	-
GSN Shi et al. (2018)	66.38	57.92	56.84	60.38
C-GCN Y. Zhang et al. (2018)	65.55	62.98	55.91	61.48
AGGCN Guo et al. (2019)	63.47	59.70	56.50	59.89
SACNN V.-H. Tran et al. (2019)	65.06	61.71	59.82	62.20
DRPC Veyseh et al. (2019)	67.30	64.28	60.19	63.92
CEON-LSTM (ours)	**68.55**	**65.42**	**61.93**	**65.30**

Table 10. F1 scores of the models on the ACE 2005 test datasets using the `word2vec` word embeddings.

Table 10. The first observation from the table is that the deep structured-based models (e.g., C-GCN, DRPC) are generally better than the deep sequence-based models (e.g., CNN, Bi-GRU) and the feature base models with large performance gaps. This demonstrates the benefits of the syntactic structures that can provide useful information to improve the performance for the deep learning models for RE. We will thus focus on these deep structure-based models in the following experiments. Among all the models, we see that the proposed model CEON-LSTM is significantly better than all the baseline models over different test domains/datasets. In particular, CEON-LSTM is 1.38% and 3.1% better than DRPC and SACNN (respectively) on the average F1 scores over different test datasets. These performance improvements are significant with $p < 0.01$ and clearly demonstrate the effectiveness of the proposed CEON-LSTM model for RE.

In order to further compare CEON-LSTM with the baselines, Table 11 presents the performance of the models when the words are represented by the

System	bc	cts	wl	Avg.
C-GCN Y. Zhang et al. (2018)	67.02	64.4	58.92	63.44
AGGCN Guo et al. (2019)	65.29	63.65	60.35	63.09
SACNN V.-H. Tran et al. (2019)	68.52	64.21	62.19	64.97
DRPC Veyseh et al. (2019)	69.41	65.82	61.65	65.62
EA-BERT X. Wang, Han, et al. (2019a)	69.25	61.70	58.48	63.14
CEON-LSTM (ours)	**71.58**	**66.92**	**65.17**	**67.89**

Table 11. F1 scores of the models on the ACE 2005 test datasets using the BERT word embeddings.

contextualized word embeddings (i.e., BERT). For this case, we also report the performance of the recent BERT-based model (i.e., Entity-Aware BERT (EA-BERT)) in X. Wang, Han, Liu, Sun, and Li (2019a) for RE on the ACE 2005 dataset. Comparing the models in Table 11 with the counterparts in 10, it is clear that the contextualized word embeddings can significantly improve the deep structure-based models for RE. More importantly, similar to the case with `word2vec`, we see that the proposed model CEON-LSTM still significantly outperforms all the baselines models with large performance gaps and $p < 0.01$, further testifying to the benefits of the CEON-LSTM model in this work.

Finally, in order to demonstrate the generalization of the proposed model over the other datasets, we show the performance of the models on the two other datasets in this work (i.e., SPOUSE and SciERC) using either `word2vec` or BERT as the word embeddings in Table 12. The results clearly confirm the effectiveness of CEON-LSTM as it is significantly better than all the other models over different datasets and word embedding settings.

3.1.3.3 Ablation Study. The Effect of the Model Components

Components: There are three major components in the proposed model: (1) the introduction of the overall sentence representation x'_t into the ON-LSTM

System	SPOUSE	SciERC
C-GCN (`word2vec`) Y. Zhang et al. (2018)	73.52	65.30
AGGCN (`word2vec`) Guo et al. (2019)	73.51	67.91
SACNN (`word2vec`) V.-H. Tran et al. (2019)	72.88	67.54
DRPC (`word2vec`) Veyseh et al. (2019)	74.66	68.18
CEON-LSTM (`word2vec`) (ours)	76.43	69.92
C-GCN (BERT) Y. Zhang et al. (2018)	75.18	74.11
AGGCN (BERT) Guo et al. (2019)	76.91	75.77
SACNN (BERT) V.-H. Tran et al. (2019)	77.98	76.42
DRPC (BERT) Veyseh et al. (2019)	78.93	77.21
CEON-LSTM (BERT) (ours)	**81.01**	**78.24**

Table 12. F1 scores of the models on the SPOUSE and SciERC datasets.

cells (called **SCG** – Sentence Context for Gates), (2) the consistency constraint for the syntax-based and model-based importance scores (called **SMC** – Syntax-Semantic Consistency), and (3) the similarity constraint for the representation vectors of the overall sentence and the shortest dependency path (called **SDPS** – Sentence-Dependency Path Similarity). In order to evaluate the contribution of these components for the overall model CEON-LSTM, we incrementally remove these components from CEON-LSTM and evaluate the performance of the remaining model. Table 13 reports the performance of the models on the ACE 2005 development dataset.

It is clear from the table that all the components are necessary for the proposed model as excluding any of them would hurt the performance significantly. It is also evident that removing more components results in more performance drop, thus demonstrating the complementary nature of the three proposed components in this work.

The Variants for CEON-LSTM: We study several variants of **SCG**, **SMC**, and **SDPS** in CEON-LSTM to demonstrate the effectiveness of the designed mechanisms. In particular, we consider the following alternatives for CEON-LSTM:

System	P	R	F1
CEON-LSTM (Full)	**74.51**	**67.29**	**71.08**
- SCG	74.00	66.98	70.45
- SMC	72.87	66.85	69.89
- SDPS	73.02	66.00	69.18
- SCG - SMC	71.52	64.62	68.08
- SCG - SDPS	70.33	64.22	67.17
- SMC - SDPS	71.02	63.95	67.58
- SCG - SMC - SDPS	70.51	63.01	66.98

Table 13. Ablation study on the development set of ACE 2005. The components listed in each row are removed from the overall model.

(i) Bi-ON-LSTM: Instead of employing the attention-based representation vectors x'_t to capture the context of the entire input sentence for the model-based importance scores in **SCG**, we run two unidirectional ON-LSTM models (i.e., the forward and backward ON-LSTM) to compute the forward and backward importance scores for each word in W. The final model-based importance score for each word is then the average of the corresponding forward and backward scores.

(ii) SA-ON-LSTM: In this method, instead of using the hidden vector h_{t-1} as the query vector to compute the attention weight α_{ti} in Equation 3.3 for **SCG**, we utilize the input representation vector x_t for w_t as the query vector (i.e., replace h_{t-1} with x_t in Equation 3.3). Consequently, SA-ON-LSTM is basically a composed model where we first run the self-attention (SA) model Vaswani et al. (2017b) over X. The results are then fed into ON-LSTM to obtain the model-based importance scores mod_t.

(iii) CE-LSTM: This aims to explore the effectiveness of ON-LSTM for our model. In CE-LSTM, we replace the ON-LSTM network with the usual LSTM model in CEON-LSTM. The **SMC** component is not included in this case as the LSTM model cannot infer the importance scores.

System	P	R	F1
CEON-LSTM (proposed)	**74.51**	**67.29**	**71.08**
Bi-ON-LSTM	72.65	67.17	69.28
SA-ON-LSTM	73.21	67.31	70.13
CE-LSTM	71.58	64.19	67.92
EP-ON-LSTM	71.03	65.16	68.45
SP-CEON-LSTM (R_W in V)	73.58	66.92	70.13
SP-CEON-LSTM (R_W not in V)	72.94	65.21	69.51

Table 14. Models' performance on the development dataset of ACE 2005.

(iv) EP-ON-LSTM: Before this work, the DRPC model in Veyseh et al. (2019) has the state-of-the-art on ACE 2005. Both DRPC and CEON-LSTM apply a more general representation of the dependency trees in a deep learning model (i.e., avoid directly using the original trees to improve the generalization). To illustrate the benefit of the importance score representation for **SMC**, EP-ON-LSTM replaces the importance score representation for the dependency trees in CEON-LSTM with the dependency edge representation in DRPC. In particular, we replace the term L_{import} in the overall loss function (i.e., Equation 3.6) with the dependency edge prediction loss (using the ON-LSTM hidden vectors) in DRPC for EP-ON-LSTM.

(v) SP-CEON-LSTM: This model removes the **SDPS** component and includes the representation vector of the dependency path DP (i.e., R_{DP}) in the final representation V for relation prediction. We consider both retaining and excluding the sentence representation R_W in V in this case. This model seeks to show that the use of R_{DP} for the similarity encouragement with R_W is more effective than employing R_{DP} directly in V.

Table 14 reports the performance of these CEON-LSTM variations on the ACE 2005 development dataset. As we can see from the table, all the considered

variants have significantly worse performance than CEON-LSTM (with $p < 0.005$). This clearly helps to justify the designs of the components **SCG**, **SMC** and **SDPS** for CEON-LSTM in this work.

3.1.4 Analysis. This section analyzes the main contribution of this work which is improving the model robustness toward noises in the dependency tree by indirectly incorporating syntactic structure with semantic hierarchy. To this end, we propose a synthetic setting in which the dependency tree has some noises and compare the main model robustness toward the noise in the dependency tree with the other two main baselines C-GCN and DRPC.

In order to introduce noise into the dependency tree, we randomly choose n words which are not on the dependency path, remove the edge between the selected nodes and their parents and add an edge between the selected nodes and a randomly selected node on the dependency path. We vary n from 0 (no noise) to 3 where 3 is the highest level of noise in the noisy dependency tree.

If the proposed model is robust toward noise in the syntactic tree, i.e., dependency tree, we expect it to have smaller performance lost than the other two baselines. More specifically, the performance gap between our model and the baseline is expected to increase by increasing the level of noise, i.e., n. The results of this analysis are shown in Table 15. Increasing noise level results in performance drop in all models as expected. However, SAON-LSTM enjoys less performance lost thanks to its indirect incorporation of syntactic tree than C-GCN which means the performance gap between these two models would increase by increasing the level of noise. This Table also shows that both SAON-LSTM and DRPC are robust toward noise level $n = 1$. It strengthen the usefulness of indirectly incorporating syntactic tree into the model. However, as SAON-LSTM exploit more structural

System	n=0	n=1	n=2	n=3
SAON-LSTM	71.08	70.63	69.03	67.21
DRPC	68.79	68.52	67.18	66.92
C-GCN	66.54	65.37	63.18	59.78

Table 15. Comparison of the performance on ACE 2005 development set when increasing the level of noise in the dependency tree.

information than DRPC, via modeling the distance to root instead of pair-wise relations between words, it has higher performance result than DRPC when level of the noise is low. Although, increasing the noise would hurt the SAON-LSTM more as it is more reliable to the whole structure of the dependency tree.

Baseline for the Model-Based Importance Scores: One of the contributions in our work is to employ the gates in the cells of ON-LSTM to obtain the model-based importance scores that are then used to promote the consistency with the syntax-based importance scores (i.e., in the **SMC** component). In order to demonstrate the effectiveness of the master cell gates to obtain the model-based importance scores, we evaluate a typical baseline where the model-based importance score mod_i for $w_i \in W$ is computed directly from the hidden vector h_i of CEON-LSTM (i.e., by feeding h_i into a feed-forward neural network with *sigmoid* activation function in the end). The model-based importance scores obtained in this way then replace the importance scores from the cell gates and are used in the **SMC** component of CEON-LSTM in the usual way (i.e., via the KL divergence in L_{import}) (note that we tried the alternatives for the KL divergence in L_{import} (i.e., the mean square error and the cosine similarity between the syntax-based and model-based importance scores), but the KL divergence produced the best results for both CEON-LSTM and HIS-CEON-LSTM on the development data). The resulting model is called **HIS-CEON-LSTM**. Table 16 reports the

System	P	R	F1
CEON-LSTM (proposed)	**74.51**	**67.29**	**71.08**
HIS-CEON-LSTM	72.02	63.97	68.29

Table 16. Models' performance on the development dataset of ACE 2005.

performance of HIS-CEON-LSTM and the proposed model CEON-LSTM on the ACE 2005 development dataset. It is clear from this table that the proposed model CEON-LSTM achieves significantly better performance than HIS-CEON-LSTM (with large performance gap), thus testifying to the importance of the master gates to obtain the model-based importance scores for CEON-LSTM.

In order to provide more insights into the performance of the proposed model, we analyze examples in the test data that can be predicted correctly with the proposed model and incorrectly with the baselines. For a baseline model M (e.g., GCN, DRPC), we call the test examples that cannot be recognized by M but can be successfully predicted by the proposed model the M-failure examples. Based on our analysis, the GCN-failure examples tend to involve the syntactic/dependency structures that does not appear or are not well represented in the training data. Some examples for the GCN-failure examples are shown in Table 17. On the one hand, as GCN is directly dependent on the syntactic structures of the input sentences, it would not be able to learn effective representations for the sentences with new structures in the GCN-failure examples for RE. On the other hand, as CEON-LSTM only exploits a relaxed general form of the tree structures (i.e., the importance scores of the words), it will be able to generalize better to the new structures in the GCN-failure examples where the general tree form is still helpful to induce effective representations for RE.

For the DRPC-failure examples (their examples are presented in Table 18), we find that these examples often involve the two entity mentions of interest with long distance from each other in the input sentences. For these examples, the dependency paths between the two entity mentions tend to be very helpful or crucial for RE as they can capture the important context words (thus eliminating the irrelevant ones). This allows the models to learn effective representations to correctly predict the relations in the sentences for RE. As DRPC only retains the dependency edges in the dependency trees separately (i.e., the local tree representations), it cannot directly capture such dependency paths, thereby failing to predict the relations for the DRPC-failure examples with long distances between the entities. This is in contrast to CEON-LSTM that exploits the global representations of the trees with the importance scores based on the distances of the words to the dependency paths. As the dependency paths can be still inferred in this global representation, CEON-LSTM can benefit from this information to successfully perform RE for the sentences in the DRPC-failure examples.

3.1.5 Conclusion. We introduce a new deep learning model for RE (i.e., CEON-LSTM) that features three major proposals. First, we represent the dependency trees via the syntax-based importance scores for the words in the input sentences for RE. Second, we propose to incorporate the overall sentence representation vectors into the cells of ON-LSTM, allowing it to compute the model-based importance scores more effectively. We also devise a novel mechanism to project the syntactic information into the computation of ON-LSTM via promoting the consistency between the syntax-based and model-based importance scores. Finally, we present a novel inductive bias for the deep learning models that exploits the similarity of the representation vectors for the whole input

Sentence	Relation
Some **Arab countries** also want to play a role in the stability operation in Iraq but are reluctant to send **troops** because of political, religious and ethnic considerations, the official said.	ORG-AFF
Some suggested that **Russian President Vladimir Putin** will now be scrambling to contain the damage to his once -budding friendship with US President **George W. Bush** because he was poorly advised by his intelligence and defense aides.	PER-SOC
Other **countries** including the Philippines, South Korea, Qatar and Australia agreed to send other help such as field **hospitals**, engineers, explosive ordnance disposal teams or nuclear, biological and chemical weapons experts.	PART-WHOLE

Table 17. The GCN-failure examples. The two entity mentions of interest are shown in bold in the sentences.

sentences and the shortest dependency paths between the two entity mentions for RE. Extensive experiments are conducted to demonstrate the benefits of the proposed model. We achieve the state-of-the-art performance on three datasets for RE. In the future, we plan to apply CEON-LSTM to other related NLP tasks (e.g., Event Extraction, Semantic Role Labeling) T. H. Nguyen, Cho, and Grishman (2016a); T. H. Nguyen and Grishman (2018a).

3.2 Multi-view Consistency for Relation Extraction via Mutual Information and Structure Prediction

In the previous section, we introduced a novel method for inferring structure for RE using Ordered-Neuron LSTM. In this section, we provide further research

Sentence	Relation
US diplomats have hinted in recent weeks that Washington 's anger with **European** resistance to the campaign was focused more on Paris –and to a lesser extent Berlin– than it was with **Moscow**.	PART-WHOLE
In **Montreal**, "Stop the War" a coalition of more than 190 groups, said as many as 200,000 **people** turned out, though police refused to give a figure.	PHYS
Although the **crossing** has, in principle, been open for movement between the two territories –while being frequently closed by **Israeli** for reasons rarely explained– the Palestinian section has been manned by Israel for more than two years.	ART

Table 18. The DRPC-failure examples. The two entity mentions of interest are shown in bold in the sentences.

on how the inferred structure can be enforced to be consistent with other structural information such as semantics-based graphs. In particular, we will show a novel application of Mutual Information for enforcing consistency between different structures inferred for RE.

The early methods for RE have involved the feature-based approach and the kernel-based approach Zelenko et al. (2003); G. Zhou et al. (2005b) where extensive feature engineering is necessary to produce effective RE models. Recently, the research in RE has essentially transformed from such feature engineering approaches to the deep learning models that have helped to significantly advance our performance on the RE benchmark datasets. Among different techniques introduced by deep learning for RE, the syntactic tress (i.e., dependency trees and constituent trees) have been shown to be a useful resource to impose the

structures over the computational graphs of the network architectures Socher, Huval, Manning, and Ng (2012); Xu et al. (2015a); Y. Zhang et al. (2018). The major benefits of such syntactic structures involve the incorporation of the semantic/syntactic hierarchies in the sentence representations Socher et al. (2012) and the capacity to directly capture the important context words for RE (i.e., via the dependency paths) Y. Zhang et al. (2018).

Despite such advantages of the syntactic structures for the deep learning models for RE, a major limitation in this approach is the reliance on some high-quality external parsers to produce the effective parse trees for the models. There are at least three issues originated from this parser reliance. First, the high-quality external parsers might only be available for some domains and languages, thus restricting the application of the RE models to those specific scenarios in practice. Second, as the external parsers are often trained only for the parsing purposes, the tree structures provided by these syntactic parsers might not be the optimal ones for RE, calling for more customized or task-specific structures for the RE problem. In fact, the syntactic trees generated by the external parsers cannot be used directly by the recent deep learning models for RE, necessitating some additional post-processing or controling mechanisms (i.e., pruning and graph attention) Guo et al. (2019); Y. Zhang et al. (2018). Finally, the external parsers are often trained for some general or specific domains for which their performance might degrade if they are applied to new domains (i.e., the domain shift problem) Plank (2011). Such performance loss of the parsers on the new domains might eventually propagate to the RE models that critically depends on the quality of the tree structures for the sentences to perform well.

In order to overcome the reliance on the external parsers, in this work, we propose to learn the implicit structures for the input sentences along with the relationship prediction for the entity metions for RE. This would help to avoid the external parsers for RE, making it possible to apply the RE models for different domains and languages, providing the task-specific sentence structures for RE, and potentially improving the RE performance on new domains. To this goal, we propose to learn the sentence structures for RE by applying two different methods to generate the dependencies/hierarchies between the words in the sentences (i.e., the two views). We would then introduce the constraints between the structures and representations learned by these two views to promote their consistencies. Our expectation is that such consistency constraints, once enforced under the relation prediction task in RE, can help to reveal the effective task-specific structures and representations for RE, a property that is impossible if the structures are pre-determined with some external parsers.

In particular, the two views for inducing the structures of the sentences involve Ordered Neurons Long-short Term Memory (ON-LSTM) Shen, Tan, Sordoni, and Courville (2019b) and self-attention Vaswani et al. (2017b) (i.e., Transformers). ON-LSTM is an extended version of the popular Long-short Term Memory networks (LSTM) that, via its internal forget and input mechanisms, can assign an importance score for each word in the sentence. Such importance scores indicate how close the words should be to the root of the tree structure induced by On-LSTM for the input sentence (i.e., the root would have highest importance score), thus implicitly forming a tree structure for the sentence. In order to perform its computation, ON-LSTM manipulates the life cycle of each neurons so the high-ranking neurons would be maintained for a larger number of steps (words) while

the low-ranking neurons would be discarded more rapidly. In contrast to the word order schema for structure induction in ON-LSTM, the seft-attention mechanism induces the structure for an input sentence by estimating the connection scores between every pair of words in the sentence (i.e., a fully connected graph structure). The score for one pair of word reflects how influential one word is with respect to the semantic understanding for the other word. Finally, in order to encourage the structure consistency between ON-LSTM and seft-attention, we transforms the pairwise scores between the words in self-attention into the importance scores for the words and promote the similarity between the two importance score sequences (i.e., from ON-LSTM and self-attention) using the KL divergence in the loss function.

In the baseline version of the aforementioned similarity promotion, we consider the importance scores of every word in the input sentence for both ON-LSTM and self-attention. However, for RE, it is possible that only a subset of the words in the sentence are necessary to correctly recognize the relationships for the entity mentions. It is thus desirable to perform the similarity promotion only on the importance scores of those relevant context words to avoid any potential noise. Consequently, in this work, we propose a filtering technique that predicts which context words in the sentence are relevant for the RE problem, and incorporates such information into the similarity promotion process to further improve the induced structures for RE.

A potential issue with the word representations induced by ON-STM and self-attention is that such word representations are excessively constrained to achieve the importance score similarity for the structure consistency, thereby loosing the important semantic information for RE. In order to alleviate this

problem, in addition to the structure consistency, we also introduce the constraints to preserve the important semantic information in the representation vectors produced by ON-LSTM and self-attention. In particular, we first use a bidirectional LSTM (BiLSTM) model to encode the semantic representations of the words in the its hidden vectors. We would then enrich the semantic content of the hidden vectors from ON-LSTM and self-attention via the semantic consistency between those vectors. In particular, we consider two mechanisms to achieve such semantic consistencies between the hidden/representation vectors in this work. The first mechanism is inspired by the control mechanism in Veyseh et al. (2019) that retains the semantic content in the representation vectors of self-attention via the control vector computed from the BiLSTM vectors of the two entity mentions. The second mechanism (newly proposed in this work), on the other hand, exploits the mutual information (MI) between the high dimension representation vectors from ON-LSTM and BiLSTM to enable their semantic consistency.

We conduct extensive experiments on the ACE 2005 and SemEval 2010 datasets that demonstrate the effectiveness of the proposed model for RE. Our model significantly outperforms the competitive baselines and achieves the state-of-the-art performance on the datasets.

3.2.1 Related Work. The early works have used the feature or kernel based approaches for RE Chan and Roth (2010); T. H. Nguyen and Grishman (2014); T. H. Nguyen et al. (2015c); Sun et al. (2011); Zelenko et al. (2003); G. Zhou et al. (2005b). These approaches often perform extensive feature engineering and might not generalize well on challenging datasets. Recently, deep learning models have been proposed to address such issues, leading to the state-of-the-art results for RE T. H. Nguyen and Grishman (2016); L. Wang et al. (2016);

D. Zeng et al. (2014); Y. Zhang et al. (2017). These deep learning models can be further categorized into the sequence based or structure based models. In the sequence based models, the sequential order of the words are preserved in the processing flow of the models (e.g., CNN T. H. Nguyen and Grishman (2015a), RNN Y. Zhang et al. (2017) or Transformer Verga et al. (2018)). In contrast, the structure-based models utilize the syntactic trees of the input sentences to structure the computational graphs of the deep learning models, thereby being able to capture the longer-term dependencies for the words. Y. Liu et al. (2015); Miwa and Bansal (2016); T. H. Nguyen and Grishman (2018a); N. Peng et al. (2017); L. Song, Zhang, Wang, and Gildea (2018); Tai et al. (2015); Xu et al. (2015a); Y. Zhang et al. (2018). However, in practice, such syntactic trees often require post-processing step (i.e., rule-based or attention-based) to customize them for RE Guo et al. (2019); Y. Zhang et al. (2018). Our work differs from these prior works as we introduce a new mechanism to automatically induce the structures for RE from the context. The learned structures are customized for RE and do not require post-processing steps.

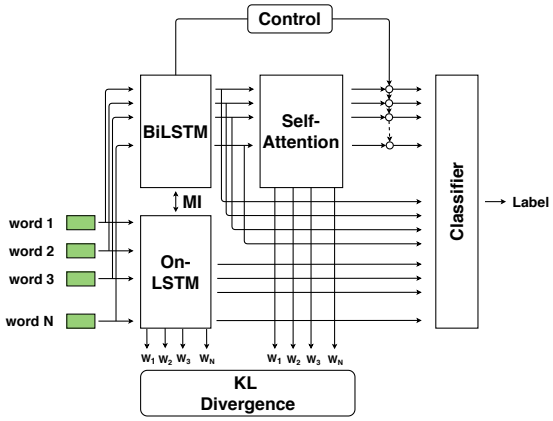

Figure 2. Model overview. The green vectors represent input word representations while the circles indicate the element-wise product.

3.2.2 Method. The RE task can be seen as a multi-class classification problem. Given a sentence $W = w_1, w_2, \ldots, w_N$ (N is the number of words/tokens in W) and two entity mentions of interest located at tokens w_s and w_o ($1 \leq s < o \leq N$), we need to predict the semantic relationship between the two entity mentions in this case.

Motivated by the prior word on RE Shi et al. (2018); Veyseh et al. (2019), we represent each word w_i in the sentence using the concatenation vector e_i of its pre-trained word embeddings, position embeddings (to indicate the positions of the two entity mentions), and entity type embeddings (to capture the entity mentions in the sentence)[3]. After this word-to-vector transformation, the input sentence W is converted into a sequence of word representation vectors $E = e_1, e_2, \ldots, e_N$ that would be used as the inputs for the next neural computations.

In order to avoid the external parsers, we propose to induce the task-specific structures during the relation prediction process for RE. Given the input sentence for RE, we employ two different network architectures to obtain the implicit structure and semantic representations for the words in the sentence. Several constraints are then introduced to ensure the consistencies between the structures and semantic representations learned by the two views. As mentioned in the introduction, the ON-LSTM and self-attention networks are used for the two views (called as the word-order view and the graph-based view respectively). Figure 2 shows the overall architecture of the proposed model. The details about the network architectures and consistency constraints are presented below.

[3]Note that different from Veyseh et al. (2019), we do not include the binary feature vectors for w_i obtained from the dependency trees (i.e., from the dependency relations and paths) as we would like to avoid the parse trees in this work.

3.2.2.1 The word-order structure view.

The strategy to induce a structure in the word-order view with ON-LSTM is to assign an importance score w_i^{onlstm} for every word w_i in the input sentence to implicitly form a binary tree structure for W. The words with higher importance scores would be closer to the root of the tree, reflecting the levels of the words in the tree. Consequently, the word w_{i*} with the highest score w_{i*}^s would be considered as the tree root, from which two subtrees are recursively constructed based on the words before w_{i*} for the left child and the words after w_{i*} for the right child.

ON-LSTM computes the importance scores for the words by introducing two additional master gates (i.e., *forget* and *input*) into the original computation of LSTM Shen et al. (2019b) (i.e., ON-LSTM is thus similar to LSTM in that both consume a sequence of vectors to produce a new sequence of hidden vectors). Basically, the hidden vectors for the *input* and *forget* gates (called the gate vectors) in both LSTM and ON-LSTM are computed for each word/step in the sentence and determine how much information of the context should be updated or forgotten respectively in the current step. However, the *forget* and *input* gate vectors in ON-LSTM differ from those in LSTM because the gates in LSTM treat every neurons/dimensions in their hidden vectors independently, imposing no hierarchy over such neurons and assuming the activity of each neuron for every word in the sentence/sequence. This is in contrast to the *forget* and *input* gates in ON-LSTM that enforce a hierarchy for the neurons/dimensions in their hidden vectors and only allow each neuron to be activated for a portion of words in the sentence. The rationale is that higher-ranking neurons would have a longer life time (i.e., being activated for more words in the sentence) to encode long-term information while

lower-ranking neurons would be canceled more rapidly to focus on the short-term information (i.e., the structural bias).

In order to achieve such ranking mechanisms for the neurons in the master gates, ON-LSTM employs the *cummax* activation function in the computation for the hidden vectors of the *forget* and *input* gates: $cumax(x) = cumsum(softmax(x))$[4]. *cummax* essentially aggregates the softmax output of some input vector x along the dimensions that can be seen as the expectation of some binary vector of the form $(0, \ldots, 0, 1, \ldots, 1)$ (i.e., divided into two consecutive segments: the 0-segment and the 1-segment). Similar to the *forget* and *input* gates in LSTM, the input for the *cummax* activation function to compute the gate vectors for ON-LSTM at the current step/word also involves the hidden vector from the previous step and the input vector for the current step. At one word/step, the 1-segments of the hidden vectors of the master gates cover the neurons that are activated for the gates at that step. Consequently, in ON-LSTM, the lengths of the 1-segments, or more precisely the sums of the weights of the neurons in the 1-segments of the gate vectors for a word are used to determine the importance of that word in the sentence. Following Shen et al. (2019b), we use the hidden vectors of the master *forget* gate in ON-LSTM to obtain the importance scores the words. In particular, let $f_i = f_{i1}, f_{i2}, \ldots, f_{iD}$ be the hidden vector for the master *forget* gate at the i-th word $w_i \in W$ from ON-LSTM (D is the dimension of the hidden vector), the importance score w_i^{onlstm} for w_i is computed by: $w_i^{onlstm} = D - \sum_{j=1..D} f_{ij}$.

In this work, we feed the input vector sequence $E = e_1, e_2, \ldots, e_N$ into two layers of ON-LSTM. We use the master *forget* gates of the second layer to generate

[4] $cumsum(u_1, u_2, \ldots, u_n) = (u'_1, u'_2, \ldots, u'_n)$ where $u'_i = \sum_{j=1..i} u_j$.

the importance sores w_i^{onlstm} for the words in W, serving as the encoding of the tree structure induced by ON-LSTM in this work. For convenience, we denote the output hidden vectors produced by the second layer of ON-LSTM for the words in the input sentence as $H' = h'_1, h'_2, \ldots, h'_N$.

3.2.2.2 The graph-based structure view. Different from the word importance scores via the number of active neurons in ON-LSTM, the graph-based structure view represents the structure for the input sentence via a fully connected graph between the words. The weight a_{ij} for the edge between w_i and w_j would indicate the level of connections/dependencies of these two words, thus implicitly defining a hierarchy among the words.

As presented previously, we obtain the graph-based structure for the input sentence in this work via the self-attention mechanisms in Transformers Vaswani et al. (2017b). In particular, starting from the input representation vectors for the words $E = e_1, e_2, \ldots, e_N$, we first feed them into a bidirectional LSTM layer (BiLSTM) that produces a sequence of hidden vectors $H = h_1, h_2, \ldots, h_N$ as the output. These representation vectors are expected to capture the semantic information for the whole input sentence. Afterward, the BiLSTM would be consumed by the self-attention layer to generate the connection scores for the pairs of words in the sentence. Specifically, for each BiLSTM vector h_i, we compute its corresponding key vector k_i, query vector q_i, and value vector v_i via: $k_i = U_k h_i$, $q_i = U_q h_i$, and $v_i = U_v h_i$ where U_k, U_q and U_v are the weight matrices, and biases are omitted for brevity in this work. The connection scores a_{ij} between the words w_i and w_j would then be obtained by the dot product between k_i and q_j:

$$a_{i,j} = \exp(k_i \cdot q_j) / \sum_{t=1..N} \exp(k_i \cdot q_t) \tag{3.7}$$

We omit the normalization factor in this formula in Vaswani et al. (2017b) for brevity. The connection scores a_{ij} of the graph structure induced by self-attention for the input sentence can be exploited for two purposes. First, they can be used to compute more abstract representation vectors $H'' = h''_1, h''_2, \ldots, h''_N$ for the words in the sentence via: $h''_i = \sum_{j=1..n} a_{ij} v_j$. Such vectors are expected to encode richer context information for the input sentence with a greater focus on the induced graph structure information. Second, the connection scores a_{ij} can also be utilized to transform the induced graph structure into a tree structure by computing an importance score for every word in the sentence and following the same strategy to generate the binary tree as we do for the ON-LSTM model. In order to obtain the importance scores for the words with a_{ij}, we assume that a word would be more important if it has stronger connections with the other words. In particular, we compute the importance score w_i^{satt} for the word w_i by:

$$w_i^{satt} = \Sigma_{j=1}^{N} a_{i,j}/N \tag{3.8}$$

where the weights $a_{i,i}$ are set to be zero. We consider the scores $w_1^{satt}, w_2^{satt}, \ldots, w_N^{satt}$ as the encoding for the tree structure learned by seft-attention in this work.

3.2.2.3 Structure consistency between views. As we do not have any supervision about the structure of the input sentence in this work, we seek to induce such structure automatically by promoting the similarity between the structures learned by the word-order and graph-based views. Intuitively, the word-order and graph-based structures should be similar/related as they are computed for the same input sentence W. We expect that such structure similarity enforcement would help to reveal the effective structures for RE. In order to achieve such structure similarity, we first transform the structure-

encoding scores $w_1^{onlstm}, w_2^{onlstm}, \ldots, w_N^{onlstm}$ and $w_1^{satt}, w_2^{satt}, \ldots, w_N^{satt}$ from ON-LSTM and self-attention into the probability distributions W^{onlstm} and W^{satt} (respectively) via: $W^{onlstm} = softmax([w_1^{onlstm}, w_2^{onlstm}, \ldots, w_N^{onlstm}])$ and $W^{satt} = softmax([w_1^{satt}, w_2^{satt}, \ldots, w_N^{satt}])$. We would then incorporate the KL divergence between W^{onlstm} and W^{satt} into the overall loss function to minimize the distance between the two distribution:

$$KL(W^{onlstm}||W^{satt}) = -\Sigma_i W_i^{onlstm} log \frac{W_i^{onlstm}}{W_i^{satt}} \tag{3.9}$$

One potential issue with Equation 3.9 is that it enforces the structure similarity based on the importance scores for the words in the sentence equally (i.e., imposing the same weight for the word-specific term in the KL divergence). This implicitly assumes the equal contribution of the words for the structure of the sentence for RE. However, in RE, there might be only a subset of words in the sentence that are actually relevant or necessary for the semantic prediction (e.g., the words along the dependency path between the two entity mentions). This suggests a mechanism where the words are weighted differently in the KL divergence for the structure consistency based on their potential contribution for the relationship prediction of the two entity mentions. Consequently, we seek to estimate a contribution score s_i for each word w_i in the KL divergence based on the BiLSTM vectors h_i, h_s and h_o of w_i and the two entity mentions of interest:

$$s_i = \sigma(W_1 \sigma(W_2[h_i, h_s, h_o])) \tag{3.10}$$

where W_1 and W_2 are the weight matrices. Finally, we use such contribution scores to weight the word-specific terms in the KL divergence in Equation 3.9 in the overall loss:

$$L_{structure} = -\Sigma_i s_i W_i^{onlstm} log \frac{W_i^{onlstm}}{W_i^{satt}} \tag{3.11}$$

3.2.2.4 Semantic consistency between views.

Given the representation vectors learned ON-LSTM (i.e., $H' = h'_1, h'_2, \ldots, h'_N$) and self-attention (i.e., $H'' = h''_1, h''_2, \ldots, h''_N$), the natural approach to perform RE is to aggregate such representation vectors to create an overall feature vector for classification. However, the structure consistency constraint in Equation 3.11 might have filtered the information content in H' and H'' excessively to encode mostly the structure information, erasing the important semantic information for RE. In order to enrich the representation vectors H' and H'' with the semantic information, we propose to distill the semantic information from the BiLSTM vectors (i.e., $H = h_1, h_2, \ldots, h_N$) and directly incorporate it into H' and H'' for RE. As the representation vectors in H are less involved with the structure constraint, we expect that they can still preserve important semantic information for RE in this case. In particular, we seek to employ mechanisms to promote the semantic consistencies between the BiLSTM representation vectors H and those from ON-LSTM and self-attention (i.e., H' and H''). We expect that such semantic consistencies can help to enrich the semantic information in the representation vectors H' and H''.

First, for the semantic consistency between H and H', we propose to maximize the mutual information (MI) between their aggregated representations in the loss function. In information theory, MI evaluates how much information we know about one random variable if the value of another variable is revealed. Two random variables would be more dependent if they have larger mutual information. Consequently, if the semantic representations from H and H' are encouraged to have large mutual information, we expect them to share more semantic information. As H already encodes the important semantic information for RE, this would help

to enrich the semantic content in H' as a by-product. In order to introduce this mutual information constraint, we first aggregate the representation vectors in H and H' into the overall representation vectors $\bar{h}$ and $\bar{h}'$ via the max-pooling function: $\bar{h} = Max\_Pooling(h_1, h_2, \ldots, h_N)$ and $\bar{h}' = Max\_Pooling(h'_1, h'_2, \ldots, h'_N)$. The mutual information would be computed between $\bar{h}$ and $\bar{h}'$ and introduced directly into the loss function for optimization.

One issue with this approach is that the computation of the MI for such high dimensional continuous vectors as $\bar{h}$ and $\bar{h}'$ is prohibitively expensive. In this work, we propose to address this issue by employing the mutual information neural estimation (MINE) in Belghazi et al. (2018) that seeks to estimate the lower bound of the mutual information between the high dimensional vectors via adversarial training. To this goal, MINE attempts to compute the lower bound of the KL divergence between the joint and marginal distributions of the given high dimensional vectors/variables. Recently it has been shown that the Jensen-Shannon divergence can also used for this purpose, offering simpler methods to compute the lower bound for the MI. Consequently, following such methods, we apply the adversarial approach to obtain the MI lower bound via the binary cross entropy of a variable discriminator. This discriminator differentiates the vectors that are sampled from the joint distribution from those that are sampled from the product of the marginal distribution of the variables. In our case, the two variables are the semantic representations $\bar{h}$ and $\bar{h}'$. In order to sample from their joint distribution, we simply concatenate $\bar{h}$ and $\bar{h}'$ (i.e., the positive example). To sample from the product of the marginal distribution, we concatenate the representation $\bar{h}$ with $\hat{h}'$ where $\hat{h}'$ is the aggregated vector (with max-pooling) of the ON-LSTM representation vectors from another sentence in the same batch with the current

sentence of interest W (i.e., the negative example). These samples are fed into a 2-layer feed forward neural network D (i.e., the discriminator) to perform a binary classification (i.e., coming from the joint distribution or the product of the marginal distributions). Finally, we use the following binary cross entropy loss to estimate the mutual information between $\bar{h}$ and $\bar{h}'$ to add into the overall loss function:

$$L_{disc} = -(log(D[\bar{h}, \bar{h}']) + log(1 - D([\bar{h}, \hat{h}'])))) \tag{3.12}$$

Second, regarding H and H'', we apply the control mechanism proposed in Veyseh et al. (2019) to enforce the semantic consistency for these representation vectors. This control mechanism first obtains a control vector c from the representation vectors in H, emphasizing on the representation vectors of the two entity mentions h_s and h_o. This control vector is then applied directly to the representation vectors in H'', obtaining a new vector $\bar{h}''_i$ for each vector $h''_i \in H''$: $\bar{h}''_i = c \odot h''_i$.

For convenience, we use $\bar{h}''$ to denote the max-pooling aggregation vector for the representation vectors $\bar{h}''_i$: $\bar{h}'' = Max\_Pooling(\bar{h}''_1, \bar{h}''_2, \ldots, \bar{h}''_N)$. Due to the direct incorporation, we expect that the semantic information in H'' would be consistent with those in H, thereby enriching the semantic content in H''. The control mechanism has been shown to work well for RE in Veyseh et al. (2019). In the experiments, we will evaluate whether we can use the control mechanism and the new MI constraint interchangeably for the semantic consistency between H, H' and H''.

3.2.2.5 Training. Finally, in order to predict the relationships between the two entity mentions w_s and w_o, we combine the representation vectors produced by BiLSTM, ON-LSTM, and self-attention to obtain an overall representation vector R for the input sentence. This representation vector involves

the max-pooling aggregation vectors from these three components (i.e., $\bar{h}$, $\bar{h}'$, and $\bar{h}''$) as well as their specific elements for the two entity mentions (i.e., h_s, h_o, h'_s, h'_o, h''_s and h''_o): $R = [\bar{h}, \bar{h}', \bar{h}'', h_s, h_o, h'_s, h'_o, h''_s, h''_o]$. Due to the structure and semantic consistencies introduced in this work, we expect R would contain effective information for the RE problem. In the final step, R would be fed into a 2 layer feed-forward neural network followed by a softmax layer to compute the probability distribution $P(.|W, s, o)$ over the possible relations for RE. We use the negative log-likelihood as the training loss in this work: $L_{pred} = -P(y|W, s, o)$ where y is the true relation label for the input sentence.

Overall, the loss function to train the model is:

$$L = L_{pred} + \alpha L_{disc} + \beta L_{structure} \tag{3.13}$$

where α and β are the trade-off parameters. The rest of this section continues with the experiments on the proposed method.

3.2.3 Experiments.

3.2.3.1 Datasets and Hyper-Parameters. We employ two widely used datasets (i.e., ACE 2005 and SemEval 2010) to evaluate the model in this work. For the ACE 2005 dataset, similar to the previous work L. Fu et al. (2017); Shi et al. (2018); Veyseh et al. (2019), we use the dataset preprocessed and provided by M. Yu et al. (2015) for compatible comparison. There are 6 different domains in this dataset, i.e., (`bc`, `bn`, `cts`, `nw`, `un`, and `wl`), covering text from news, conversations and web blogs. Following the the prior work, the union of the domains `bn` and `nw` (called `news`) is used as the training data (called the source domain); a half of the documents in `bc` is reserved for the development data, and the remainder (`cts`, `wl` and the other half of `bc`) serve as the test data (called the target domains). This data separation helps to evaluate the cross-domain

System	bc	cts	wl	Avg.
FCM M. Yu et al. (2015)	61.90	52.93	50.36	55.06
Hybrid FCM M. Yu et al. (2015)	63.48	56.12	55.17	58.25
LRFCM M. Yu et al. (2015)	59.40	-	-	-
Log-linear T. H. Nguyen and Grishman (2016)	57.83	53.14	53.06	54.67
CNN T. H. Nguyen and Grishman (2016)	63.26	55.63	53.91	57.60
Bi-GRU T. H. Nguyen and Grishman (2016)	63.07	56.47	53.65	57.73
Forward GRU T. H. Nguyen and Grishman (2016)	61.44	54.93	55.10	57.15
Backward GRU T. H. Nguyen and Grishman (2016)	60.82	56.03	51.78	56.21
CNN+DANN L. Fu et al. (2017)	65.16	-	-	-
GSN Shi et al. (2018)	66.38	57.92	56.84	60.38
AGGCN Guo et al. (2019)	63.47	59.70	56.50	59.89
SACNN V.-H. Tran et al. (2019)	65.06	61.71	59.82	62.20
DRPC Veyseh et al. (2019)	67.30	64.28	60.19	63.92
MVC (ours)	**70.32**	**66.43**	**64.61**	**68.20**

Table 19. F1 scores of the models on the ACE 2005 dataset over different target domains `bc`, `cts`, and `wl`.

generalization of the models due to the domain difference of the training data and test data. For the SemEval 2010 dataset Hendrickx et al. (2010), there are 18 semantic relations that along with an *Other* class, leading to a 19-class classification problem. As validation data is not provided in SemEval 2018, we use the same model parameters as those used for the ACE 2005 dataset for consistency.

Based on the fine-tuning process on the validation data of the ACE 2005 dataset, we find the following values for the hyper-parameters for the proposed model: 50 dimensions for the position embeddings and entity type embeddings, 100 hidden units for the BiLSTM and ON-LSTM models, 200 dimensions for all the other hidden vectors in the model (i.e., the hidden vectors in self-attention and the layers of the feed-forward neural networks), 0.1 for the loss trade-off parameters α and β, and 0.3 for the learning rate with the Adam optimizer. Finally, we use the pre-trained word embedding `word2vec` with 300 dimension to represent the words.

3.2.3.2 Comparison to the state of the art.

We compare the performance of the proposed model (called MVC for multi-view consistency) with the following baselines:

• Feature based models: These models use linguistic features for RE, i.e., FCM, Hybrid FCM, LRFCM, and SVM Hendrickx et al. (2010); M. Yu et al. (2015).

• Deep sequential models: These models employ deep learning architectures based on the sequential order of the sentence for RE, i.e., log-linear, CNN, Bi-GRU, Forward GRU, Backward GRU T. H. Nguyen and Grishman (2016), and CNN+DANN L. Fu et al. (2017).

• Adversarial learning model: This model, called GSN, is trained to learn the genre agnostic features for cross-domain RE. Shi et al. (2018)

• Deep structure-based models: These models employ dependency trees either as the input features or graphs to form the computation flow for deep learning models. The state-of-the-art models of this type include: AGGCN Guo et al. (2019), SACNN V.-H. Tran et al. (2019) and DRPC Veyseh et al. (2019). DRPC has the best reported performance on ACE 2005. Note that we obtain the performance of these models on the considered datasets using the actual implementation released by the original papers.

We report the F1 scores on all the ACE 2005 test sets in Table 19. From the table, we see that the deep structure-based models (i.e., AGGCN, SACNN and DRPC) are in general better than the deep sequential models, thus suggesting the benefits of the sentence structures (i.e., the dependency trees from the external parsers) for deep learning for RE. More importantly, the proposed model MVC is shown to significantly outperforms the baseline models on all the test sets with

System	F1
SVM Hendrickx et al. (2010)	82.2
SDP-LSTM Xu et al. (2015a)	83.7
SPTree Miwa and Bansal (2016)	84.4
PA-Tree Y. Zhang et al. (2017)	82.7
C-GCN Y. Zhang et al. (2018)	84.8
LISA Strubell, Verga, Andor, Weiss, and McCallum (2018)	83.9
DRPC Veyseh et al. (2019)	85.2
AGGCN Guo et al. (2019)	85.7
SACNN V.-H. Tran et al. (2019)	85.8
MVC (ours)	**86.1**

Table 20. Performance on the SemEval 2010 dataset.

$p < 0.01$. The performance gap is substantial and clearly demonstrates the effectiveness of the proposed MVC in this work. In particular, MVC improves the average F1 score of the deep sequential models by almost 10% while this performance improvement for the structure-based model is at least 4%. We attribute the better performance of MVC over the structure-based models to the fact that the task-specific and context-dependent structures from MVC is better suited for RE than the pre-defined structures from the external parsers. Finally, due to the cross-domain nature of the evaluation on the ACE 2005 dataset, we can also conclude that the task-specific structures learned by MVC can be more robust against the domain shifts for RE.

We also compare the performance of MVC (i.e., using the macro F1 score) with the state-of-the-art structured-based RE models (i.e., using dependency trees) on the SemEval 2010 test set in Table 20. These models are also selected for comparison in Veyseh et al. (2019). As we can see from the table, MVC can achieve significantly better or comparable performance with the other structure-based methods, further testifying to the advantages of the task-specific structure induction for RE proposed in this work.

3.2.3.3 Ablation study on components.

In this section, we report the performance of the proposed model on the ACE 2005 development set when the major components of the model is excluded. In particular, we seek to evaluate the contribution of four main components in this work, including the BiLSTM module, the ON-LSTM module, the self-attention module, and the contribution scores s_i for the KL divergence constraint in Equation 3.10. The results are shown in Table 21. As we can see from the table, all the components are necessary for MVC as removing any of them would hurt the performance significantly. The largest performance loss comes from excluding ON-LSTM that highlights the importance of ON-LSTM on inducing effective structure and semantic information for RE. Importantly, the performance loss due to the elimination of the contribution scores s_i in Equation 3.10 suggests that in RE, learning the structure throughout the entire sentences would not be as helpful as restricting the structure induction to the relevant parts of the sentences.

3.2.3.4 Semantic consistency.

There are two mechanisms for semantic consistency in this work, i.e., the MI constraint and the control mechanism Veyseh et al. (2019). In the model, the MI constraint is used to promote the semantic consistency between the representation vectors from BiLSTM and ON-LSTM (i.e., BiLSTM $\leftrightarrow$ ON-LSTM) while the control mechanism is applied for the vectors from BiLSTM and self-attention (i.e., BiLSTM $\leftrightarrow$ self-attention). The natural question is whether the MI and control mechanisms can be used interchangeably to achieve the semantic consistencies for the representation vectors for the two pairs BiLSTM $\leftrightarrow$ ON-LSTM and BiLSTM $\leftrightarrow$ self-attention. Table 22 shows the performance of the 4 possible combinations of the MI and control mechanisms for the two semantic consistency pairs BiLSTM $\leftrightarrow$ ON-

System	P	R	F1
MVC	77.8	65.1	70.1
MVC - LSTM	75.7	62.9	68.0
MVC - ON-LSTM	72.1	63.4	67.5
MVC - SA	80.1	61.2	68.4
MVC - Contribution Scores in (3.10)	73.2	66.5	69.2

Table 21. Ablation study on the ACE 2005 dev set.

Mechanisms	P	R	F1
MI, Control (proposed)	77.8	65.1	70.1
MI, MI	84.5	59.6	67.3
Control, MI	77.2	64.5	69.2
Control, Control	74.2	63.2	68.9

Table 22. Performance on the ACE 2005 dev set when the MI and control mechanisms are used interchangeably. The first and second mechanisms in each row corresponds to the constraints for BiLSTM $\leftrightarrow$ ON-LSTM and BiLSTM $\leftrightarrow$ self-attention respectively.

LSTM and BiLSTM $\leftrightarrow$ self-attention. This table shows that when we use only one type of the semantic consistency mechanisms, i.e. both MI or both control, the performance drops more than the cases with two types of mechanisms. This demonstrates the complementary effects between the MI and control constraints for semantic consistency for RE. The best performance is achieved when the MI mechanism is used for BiLSTM $\leftrightarrow$ ON-LSTM and the control mechanism is reserved for BiLSTM $\leftrightarrow$ self-attention, testifying to our design of the proposed model in this work.

3.2.3.5 Ablation study on consistency. This section investigates whether the consistency constraints (i.e., structure and semantics) are necessary. Table 23 presents the performance of MVC when different combinations of the consistency constraints are removed from the model. As we can see from the table, the model's performance would be reduced significantly if any of

Model	P	R	F1
MVC	77.8	65.1	70.1
MVC - KL	72.3	65.8	68.1
MVC - MI	74.0	67.4	69.5
MVC - Control	78.4	61.9	68.5
MVC - KL - MI	71.2	66.1	68.0
MVC - MI - Control	70.1	65.6	67.9
MVC - KL - Control	70.5	66.4	68.2
MVC - KL - MI - Control	72.1	63.2	67.1

Table 23. Performance on the ACE 2005 dev set when the consistency constraints are removed from the model.

the constraints is excluded, among which the KL divergence constraint for the structure consistency between ON-LSTM and self-attention would lead to the most significant performance loss. Importantly, when all the three constraints are excluded from MVC (thus making it an ensemble model between ON-LSTM and self-attention), the performance would become the worst (i.e., 3% loss in the F1 score). These results clearly demonstrate the effectiveness of the proposed MVC in this work, highlighting the consistency constraints as the important mechanisms to achieve good performance for RE.

3.2.4 Conclusion. We propose a novel method for RE that seeks to automatically induce the task-specific structures for the input sentences, avoiding the external parsers. The experiments show that the induced structures are more effective for RE than the pre-defined structures from the external parsers. The key innovation of the proposed method is to use two views (i.e., ON-LSTM and self-attention) to learn the structures and semantic representations for the input sentences. We introduce several constraints to enforce the structure and semantic consistencies between the two views based on KL differences and mutual information. We achieve the state-of-the-art performance for RE on both the

cross-domain and general settings, thereby demonstrating the effectiveness and the robustness of the proposed model.

3.3 Conclusion

In this Chapter, we study two novel methods based on On-LSTM and Self-Attention to infer effective structures for the task of Relation Extraction. Further, we provide details on a novel mechanism to ensure consistency between these views using Mutual Information. In the next Chapter, we take a step further and instead of inferring structure between words in an input text, we infer structure between samples in a batch of data. Such a structure can be used to ensure the consistency of the data points in a batch.

CHAPTER IV

INFERRING STRUCTURES FOR IE AT DATASET LEVEL

This Chapter contains materials from the published paper: "*Amir Pouran Ben Veyseh, Viet Lai, Franck Dernoncourt, and Thien Huu Nguyen. '**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. 2021*". In this publication, the experiments were entirely done by the author of this dissertation, Amir Pouran Ben Veyseh. The other co-authors provided feedback regarding the experiments and results. Amir wrote the entire paper and Dr. Thien Huu Nguyen provided editorial feedback for this paper.

In previous chapters, we study novel methods for exploiting existing structures or inferring task-specific structures for IE problems. Although these structures are helpful to improve the performance of IE models, they are limited to a single input data point. In other words, the interactions between the samples have not been studied so far. Such interactions could be modeled as the similarity between data points which could be helpful to increase the generalization ability of the IE model. As such, in this Chapter, we study how the interaction between samples in a batch of data can be exploited for an IE model. In particular, the similarity between samples can be used to construct a bipartite graph that can measure the consistency between different data points in a batch of data. The remainder of this Chapter provides details of the proposed technique.

An important task of Information Extraction (IE) involves Event Detection (ED) whose goal is to recognize and classify words/phrases that evoke events in text (i.e., event triggers). For instance, in the sentence "*The organization **donated***

2 million dollars to humanitarian helps.", ED systems should recognize "*donated*" as an event trigger of type *Pay*. We differentiate two subtasks in ED, i.e., Event Identification (EI): a binary classification problem to predict if a word in text is an event trigger or not, and Event Classification (EC): a multi-class classification problem to classify event triggers according to predefined event types.

Several methods have been introduced for ED, extending from feature-based models Ahn (2006); Liao and Grishman (2010a); Miwa et al. (2014) to advanced deep learning methods Y. Chen et al. (2015); M. V. Nguyen, Lai, and Nguyen (2021a); T. H. Nguyen and Grishman (2015a); T. H. Nguyen, Meyers, and Grishman (2016g); Sha et al. (2018); Y. Zhang, Xu, et al. (2020). Although deep learning models have achieved substantial improvement, their requirement of large training datasets together with the small sizes of existing ED datasets constitutes a major hurdle to build high-performing ED models. Recently, there have been some efforts to enlarge training data for ED models by exploiting unsupervised L. Huang et al. (2016); Yuan et al. (2018) or distantly-supervised Araki and Mitamura (2018); Keith et al. (2017); M. Nguyen and Nguyen (2018b) techniques. The common strategy in these methods is to exploit unlabeled text data that are rich in event mentions to aid the expansion of training data for ED. In this work, we explore a novel approach for training data expansion in ED by leveraging the existing pre-trained language model GPT-2 Radford et al. (2019) to automatically generate training data for models. Motivated by the promising performance of GPT models for text generation, we expect our approach to produce effective data for ED in different domains.

Specifically, we aim to fine-tune GPT-2 on existing training datasets so it can generate new sentences annotated with event triggers and/or event types,

serving as additional training data for ED models. One direction to achieve this idea is to explicitly mark event triggers along with their event types in sentences of an existing ED dataset that can be used to fine-tune the GPT model for new data generation. However, one issue with this direction is that in existing ED datasets, numbers of examples for some rare event types might be small, potentially leading to the poor tuning performance of GPT and impairing the quality of generated examples for such rare events. In addition, large numbers of event types in some ED datasets might make it more challenging for the fine-tuning of GPT to differentiate event types and produce high-quality data. To this end, instead of directly generating data for ED, we propose to use GPT-2 to only generate samples for the event identification task to simplify the generation and achieve data with better annotated labels (i.e., output sentences only are only marked with positions of event triggers). As such, to effectively leverage the generated EI data to improve ED performance, we propose a multi-task learning framework to train the ED models on the combination of the generated EI data and the original ED data. In particular, for every event trigger candidate in a sentence, our framework seeks to perform two tasks, i.e., EI to predict a binary label for being an event trigger or not, and ED to predict the event type (if any) evoked by the word via a multi-class classification problem. An input encoder is shared for both tasks that allow training signals from both generated EI data and original ED data to contribute to the representation learning in the encoder (i.e., transferring knowledge in generated EI data to ED models).

Despite the simplification to EI for better annotated labels of data, the generated sentences might still involve noises due to the inherent nature of the language generation, e.g., grammatically wrong sentences, inconsistent

information, or incorrect event trigger annotations. As such, it is crucial to introduce mechanisms to filter the noises in generated data to enable effective transfer learning from generated EI data. To this end, prior works for GPT-based data generation for other tasks has attempted to directly remove noisy generated examples before actual usage for model training via some heuristic rules Anaby-Tavor et al. (2020); Y. Yang et al. (2020). However, heuristic rules are brittle and restricted in their coverage so they might overly filter the generated data or incorrectly retain some noisy generated samples. To address this issue, we propose to preserve all generated data for training and devise methods to explicitly limit impacts of noisy generated sentences in the models. In particular, we expect the inclusion of generated EI data into the training process for ED models might help to shift the representations of the models to better regions for ED. As such, we argue that this representation transition should only occur at a reasonable rate as drastic divergence of representations due to the generated data might be associated with noises in the data. Motivated by this intuition, we propose a novel teacher-student framework for our multi-task learning problem where the teacher is trained on the original clean ED datasets to induce anchor representation knowledge for data. The student, on the other hand, will be trained on both generated EI data and original ED data to accomplish transfer learning. Here, the anchor knowledge from the teacher will be leveraged to guide the student to prevent drastic divergence of representation vectors for noisy information penalization. Consequently, we propose a novel anchor information to implement this idea, seeking to maintain the same level of differences between the generated and original data (in terms of representation vectors) for both the teacher and the student (i.e., generated-vs-original data difference as the anchor). At the core of this techniques

involves the computation of distance/difference between samples in generated and original data. In this work, we envision two types of information that models should consider when computing such distances for our problem: (1) representation vectors of the models for the examples, and (2) event trigger likelihood scores of examples based on the models (i.e., two examples in the generated and original data are more similar if they both correspond to event triggers). As such, we propose to cast this distance computation problem of generated and original data into an Optimal Transport (OT) problem. OT is an established method to compute the optimal transportation between two data distributions based on the probability masses of data points and their pair-wise distances, thus facilitating the integration of the two criteria of event trigger likelihoods and representation vectors into the distance computation between data point sets.

Extensive experiments and analysis reveal the effectiveness of the proposed approach for ED in different domains, establishing new state-of-the-art performance on the ACE 2005, CySecED and RAMS datasets.

4.1 Model

We formulate the task of Event Detection as a word-level classification problem as in prior work Ngo, Nguyen, and Nguyen (2020); T. H. Nguyen and Grishman (2015a). Formally, given the sentence $S = [w_1, w_2, \ldots, w_n]$ and the candidate trigger word w_t, the goal is to predict the event type l from a pre-defined set of event types L. Note that if the word w_t is not a trigger word, the gold event type is $None$. Our proposed approach for this task consist of two stages: (1) Data Augmentation: to employ natural language generation to augment existing training datasets for ED, (2) Task Modeling: to propose a deep learning model for ED, exploiting available training data.

4.1.1 Data Augmentation. As presented in the introduction, our motivation in this work is to explore a novel approach for training data augmentation for ED based on the powerful pre-trained language model for text generation GPT2. Our overall strategy involves using some existing training dataset $\mathcal{O}$ for ED (i.e., original data) to fine-tune GPT-2. The fine-tuned model is then employed to generate a new labeled training set $\mathcal{G}$ (i.e., synthetic data) that will be combined with the original data $\mathcal{O}$ to train models for ED.

To simplify the training data generation task and enhance the quality of the synthetic data, we seek to generate data only for the subtask EI of ED where synthesized sentences are annotated with positions of their event triggers (i.e., event types for triggers are not required for the generation to avoid the complication with rare event types for fine-tuning). To this end, we first enrich each sentence $S \in \mathcal{O}$ with positions of event triggers that it contains to facilitate the GPT fine-tuning process. Formally, assume that $S = w_1, w_2, \ldots, w_n$ is a sentence of n words with only one event trigger word located at w_t, the enriched sentence S' for S would have the form: $S' = [BOS, w_1, \ldots, TRG_s, w_t, TRG_e, \ldots, w_n, EOS]$ where TRG_s and TRG_e are special tokens to mark the position of the event trigger, and BOS and EOS are special tokens to identify the beginning and the end of the sentence. Next, the GPT-2 model will be fine-tuned on the enriched sentences S' of $\mathcal{O}$ in an auto-regressive fashion (i.e., predicting the next token in S' given prior ones). Finally, using the fine-tuned GPT-2, we generate a new dataset $\mathcal{G}$ of $|\mathcal{O}|$ sentences ($|\mathcal{G}| = |\mathcal{O}|$) to achieve a balanced size. Here, we ensure that only generated sentences that contain the special tokens TRG_s and TRG_e (i.e., involving event trigger words) are added into $\mathcal{G}$, allowing us to identify the candidate trigger word in our word-level classification formulation for ED. As such, the combination

$\mathcal{A}$ of the synthetic data $\mathcal{G}$ and the original data $\mathcal{O}$ ($\mathcal{A} = \mathcal{O} \cup \mathcal{G}$) will be leveraged to train our ED model in the next step.

To assess the quality of the synthetic data, we randomly select 200 sentences from $\mathcal{G}$ (generated by the fine-tuned GPT-2 model over the popular ACE 2005 training set for ED) and evaluate them regarding grammatical soundness, meaningfulness, and inclusion and correctness of annotated event triggers (i.e., whether the words between the tokens TRG_s and TRG_e evoke events or not). Among the sampled set, we find that 17% of the sentences contains at least one type of such errors.

4.1.2 Task Modeling. This section describes our model for ED to overcome the noises in the generated data $\mathcal{G}$ for model training. As discussed in the introduction, we employ the Teacher-Student framework with multi-task learning to achieve this goal. In the proposed framework, the teacher and student employs a base deep learning model with the same architecture and different parameters.

Base Model: Following the prior work X. Wang, Han, Liu, Sun, and Li (2019b), our base model consists of the BERT$_{base}$ model to represent each word w_i in the input sentence S with a vector e_i. Formally, the input sentence $[[CLS], w_1, w_2, \ldots, w_n, [SEP]]$ is fed into the BERT$_{base}$ model and the hidden states of the last layer of BERT are taken as the contextualized embeddings of the input words, i.e., $E = [e_1, e_2, \ldots, e_n]$. Note that if w_i contains more than one word-piece, the average of its word-piece embeddings is used for e_i. In our experiments, we find that fixing the BERT$_{base}$ parameters achieve higher performance. As such, to fine-tune the contextualized embeddings E for ED, we employ a Bi-directional Long Short-Term Memory (BiLSTM) network to consumes E; its hidden states, i.e., $H = [h_1, h_2, \ldots, h_n]$, are then employed as the final representations for the

words in S. Finally, to create the final vector V for ED prediction, the max-pooled representation of the sentence, i.e., $\bar{h} = MAX\_POOL(h_1, h_2, \ldots, h_n)$, is concatenated with the representation of the trigger candidate, i.e., h_t. V is consumed by a feed-forward network, whose last layer has $|L|$ neurons, followed by a softmax layer to predict the distribution $P(\cdot|S, t)$ over possible event types in L. To train the model, we use negative log-likelihood as the loss function: $\mathcal{L}_{pred} = -\log P(l|S, t)$ where l is the gold label.

As the synthetic sentences in $\mathcal{G}$ only involve information about positions of event triggers (i.e., no event types included), we cannot directly combine $\mathcal{G}$ with $\mathcal{O}$ to train ED models with the loss $\mathcal{L}_{pred}$. To facilitate the integration of $\mathcal{G}$ into the training process, we introduce an auxiliary task of EI for the multi-task learning in the training process, seeking to predict the binary label l_{aux} for the trigger candidate w_t in S, i.e., $l_{aux} = 1$ if w_t is an event trigger. To perform this auxiliary task, we employ another feed-forward network, i.e., $\texttt{FF}_{aux}$, which also consumes the overall vector V as input. This feed-forward network has one neuron with the sigmoid activation function in the last layer to estimate the event trigger likelihood score: $P(l_{aux} = 1|S, t) = \texttt{FF}_{aux}(V)$. Finally, to train the base model with the auxiliary task, we exploit the binary cross-entropy loss: $\mathcal{L}_{aux} = -(l_{aux} \log(\texttt{FF}_{aux}(V)) + (1 - l_{aux}) \log(1 - \texttt{FF}_{aux}(V)))$. Note that the main ED task and the auxiliary EI task are done jointly in a single training process where the loss $\mathcal{L}_{pred}$ for ED is computed only for the original data $\mathcal{O}$. The loss $\mathcal{L}_{aux}$, in contrast, will be obtained for both original and synthetic data in $\mathcal{A}$.

Knowledge Consistency: The generated data $\mathcal{G}$ is not noise-free. As such, training the ED model on $\mathcal{A}$ could lead to inferior performance. To address this issue, as discussed in the introduction, we propose to first learn the anchor

knowledge from the original data $\mathcal{O}$, then use that to lead the model training on $\mathcal{A}$ to prevent drastic divergence from the anchor knowledge (i.e., knowledge consistency promotion), thus constraining the noises. Hence, we propose a teacher-student network, in which the teacher is first trained on $\mathcal{O}$ to learn the anchor knowledge. The student network will be trained on $\mathcal{A}$ afterward leveraging the consistency guidance with the induced anchor knowledge from the teacher. We will also use the student network as the final model for our ED problem in this work.

In our framework, both teacher and student networks will be trained in the multi-task setting with ED and EI tasks. In particular, the training losses for both ED and EI will be computed based on $\mathcal{O}$ for the teacher (the loss to train the teacher is: $\mathcal{L}_{pred} + \tau\mathcal{L}_{aux}$ where τ is a trade-off parameter). In contrast, the combined data $\mathcal{A}$ will be used to compute the EI loss for the student while the ED loss for the student can only be computed on the original data $\mathcal{O}$. As such, we propose to enforce the knowledge consistency between the two networks for both the main task ED and the auxiliary task EI during the training of the student model. First, to achieve the knowledge consistency for ED, we seek to minimize the KL divergence between the teacher-predicted label-probability distribution and the student-predicted label-probability distributions. Formally, for a sentence $S \in \mathcal{O}$, the label-probability distributions of the teacher and the student, i.e., $P_t(\cdot|S,t)$ and $P_s(\cdot|S,t)$ respectively, are employed to compute the KL-divergence loss $\mathcal{L}_{KL} = -\Sigma_{l\in L} P_t(l|S,t)\log(\frac{P_t(l|S,t)}{P_s(l|S,t)})$. By decreasing the KL-divergence during the student's training, the model is encouraged to make similar predictions as the teacher for the same original sentence, thereby preventing noises to mislead the student. Note that different from traditional teacher-student networks that employ KL to achieve knowledge distillation on unlabelled data Hinton, Vinyals,

and Dean (2015), the KL divergence in our model is leveraged to enforce knowledge consistency to prevent noises in labeled data automatically generated by GPT-2.

Second, for the auxiliary task EI, instead of enforcing the student-teacher knowledge consistency via similarity predictions, we argue that it will be more beneficial to leverage the difference between the original data $\mathcal{O}$ and the generated data $\mathcal{G}$ as an anchor knowledge to promote consistency. In particular, we expect that the student which is trained on $\mathcal{A}$, should discern the same difference between $\mathcal{G}$ and $\mathcal{O}$ as the teacher which is trained only on the original data $\mathcal{O}$. Formally, during student training, for each mini-batch, the distances between the original data and the generated data detected by the teacher and the student are denoted by $d^T_{\mathcal{O},\mathcal{G}}$ and $d^S_{\mathcal{O},\mathcal{G}}$, respectively. To enforce the $\mathcal{O}$-$\mathcal{G}$ distance consistency between the two networks, the following loss is added into the overall loss function: $\mathcal{L}_{dist} = \frac{|d^T_{\mathcal{O},\mathcal{G}} - d^S_{\mathcal{O},\mathcal{G}}|}{|B|}$, where $|B|$ is the mini-batch size. The advantage of this novel knowledge consistency enforcement compared to the KL-divergence is that it explicitly exploits the different nature of the original and generated data to facilitate the mitigation of noises in the generated data.

A remaining question for our proposed knowledge consistency concerns how to assess the difference between the original and the generated data from the perspective of the teacher, i.e., $d^T_{\mathcal{O},\mathcal{G}}$, and the student networks, i.e., $d^S_{\mathcal{O},\mathcal{G}}$. In this section, we will describe our method from the perspective of the student (the same method is employed for the teacher network). In particular, we define the difference between the original and the generated data as the cost of transforming $\mathcal{O}$ to $\mathcal{G}$ such that for the transformed data the model will make the same predictions as $\mathcal{G}$. How can we compute the cost of such transformation? To answer this question, we propose to employ Optimal Transport (OT) which is an established method

to find the efficient transportation (i.e., transformation with the lowest cost) of one probability distribution to another one. Formally, given the probability distributions $p(x)$ and $q(y)$ over the domains $\mathcal{X}$ and $\mathcal{Y}$, and the cost function $C(x, y) : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}_+$ for mapping $\mathcal{X}$ to $\mathcal{Y}$, OT finds the optimal joint 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:

$$\pi^*(x, y) = \min_{\pi \in \Pi(x,y)} \int_{\mathcal{Y}} \int_{\mathcal{X}} \pi(x, y) C(x, y) dx dy$$
$$\textbf{s.t. } x \sim p(x) \text{ and } y \sim q(y), \tag{4.1}$$

where $\Pi(x, y)$ is the set of all joint distributions with marginals $p(x)$ and $q(y)$. Note that if the distributions $p(x)$ and $q(y)$ are discrete, the integrals in Equation 4.1 are replaced with a sum and the joint distribution $\pi^*(x, y)$ is represented by a matrix whose entry (x, y) represents the probability of transforming the data point $x \in \mathcal{X}$ to $y \in \mathcal{Y}$ to convert the distribution $p(x)$ to $q(y)$. By solving the problem in Equation 4.1[1], the cost of transforming the discrete distribution $p(x)$ to $q(y)$ (i.e., Wasserstein distance $Dist_W$) is defined as: $Dist_W = \Sigma_{x \in \mathcal{X}} \Sigma_{y \in \mathcal{Y}} \pi^*(x, y) C(x, y)$.

In order to utilize OT to compute the transformation cost between $\mathcal{O}$ and $\mathcal{G}$, i.e., $d^S_{\mathcal{O},\mathcal{G}}$, we propose to define the domain $\mathcal{X}$ and $\mathcal{Y}$ as the representation spaces of the sentences in $\mathcal{O}$ and $\mathcal{G}$, respectively, obtained from the student network. In particular, a data point $x \in \mathcal{X}$ represents a sentence $X_o \in \mathcal{O}$. Similarly, a data point $y \in \mathcal{Y}$ stands for a sentence $Y_g \in \mathcal{G}$. To define the cost function $C(x, y)$ for OT, we compute the Euclidean distance between the representation vectors of the sentences X_o and Y_g (obtained by max-pooling over representations of their words): $C(x, y) = \left\| \bar{h}^X_o - \bar{h}^Y_g \right\|$ where $\bar{h}^X_o = MAX\_POOL(h^X_{o,1}, \ldots, h^X_{o,|X_o|})$, $\bar{h}^Y_g = MAX\_POOL(h^Y_{g,1}, \ldots, h^Y_{g,|Y_g|})$, and $h^X_{o,i}$ and $h^Y_{g,i}$ are the representation vectors of

[1]It is worth mentioning that this problem is intractable so we solve its entropy-based approximation using the Sinkhorn algorithm Peyre and Cuturi (2019).

the i-th words of X_o and Y_g, respectively, obtained from the student's BiLSTM. Also, to define the discrete distribution $p(x)$ for OT over $\mathcal{X}$, we employ the event trigger likelihood $Score_o^X$ for the trigger candidate of each sentence X_o in $\mathcal{X}$ that is returned by the feed-forward network $\texttt{FF}_{aux}^S$ for the auxiliary task EI in the student model, i.e, $Score_o^X = \texttt{FF}_{aux}^S(X_o)$. Afterward, we apply the softmax function over the scores of the original sentences in the current mini-batch to obtain $p(x)$, i.e., $p(x) = Softmax(Score_o^X)$. Similarly, the discrete distribution $q(y)$ is defined as $q(y) = Softmax(Score_g^Y)$. To this end, by solving the OT problem in Equation 4.1 and obtaining the efficient transport plan $\pi^*(x, y)$ using this setup, we can obtain the distance $d_{\mathcal{O},\mathcal{G}}^S$. In the same way, the distance $d_{\mathcal{O},\mathcal{G}}^T$ can be computed using the representations and event trigger likelihoods from the teacher network. Note that in this way, we can integrate both representation vectors of sentences and event trigger likelihoods into the distance computation between data as motivated in the introduction.

Finally, to train the student model, the following combined loss function is used in our framework: $\mathcal{L} = \mathcal{L}_{pred} + \alpha\mathcal{L}_{aux} + \beta\mathcal{L}_{KL} + \gamma\mathcal{L}_{dist}$, where α, β, and γ are the trade-off parameters.

4.2 Experiments

4.2.1 Datasets, Baselines & Hyper-Parameters. To evaluate the effectiveness of the proposed model, called the GPT-based data augmentation model for ED with OT (GPTEDOT), we conduct experiments on the following ED datasets:

ACE 2005 Walker, Strassel, Medero, and Maeda (2006b): This dataset annotates 599 documents for 33 event types that cover different text domains(e.g., news, weblog or conversation documents). We use the same pre-processing script

and data split as prior works Lai, Nguyen, and Nguyen (2020b); Tong, Xu, et al. (2020) to achieve fair comparisons. In particular, the data split involves 529/30/40 articles for train/dev/test sets respectively. For this dataset, we compare our model with prior state-of-the-art models reported in the recent works Lai, Nguyen, and Nguyen (2020b); Tong, Xu, et al. (2020), including BERT-based models such as DMBERT, AD-DMBERT X. Wang, Han, et al. (2019b), DRMM, EKD Tong, Xu, et al. (2020), and GatedGCN Lai, Nguyen, and Nguyen (2020b).

CySecED Man Duc Trong, Trong Le, Pouran Ben Veyseh, Nguyen, and Nguyen (2020): This dataset provides 8,014 event triggers for 30 event types from 300 articles of the cybersecurity domain (i.e., cybersecurity events). We follow the the same pre-processing and data split as the original work Man Duc Trong et al. (2020) with 240/30/30 documents for the train/dev/test sets. To be consistent with other experiments and facilitate the data generation based on GPT-2, the experiments on CySecED are conducted at the sentence level where inputs for models involve sentences. As such, we employ the state-of-the-art sentence-level models reported in Man Duc Trong et al. (2020), i.e., DMBERT X. Wang, Han, et al. (2019b), BERT-ED S. Yang et al. (2019), as the baselines for CySecED.

RAMS Ebner et al. (2020): This dataset annotates 9,124 event triggers for 38 event types. We use the official data split with 3,194, 399, and 400 documents for training, development, and testing respectively for RAMS. We also perform ED at the sentence level in this dataset. For the baselines, we utilize recent state-of-the-art BERT-based models for ED, i.e., DMBERT X. Wang, Han, et al. (2019b) and GatedGCN Lai, Nguyen, and Nguyen (2020b). For a fair comparison, the performance of such baseline models is obtained via their official implementations from the original papers that are fine-tuned for RAMS.

Model	P	R	F1
CNN T. H. Nguyen and Grishman (2015a)	71.8	66.4	69.0
DMCNN Y. Chen et al. (2015)	75.6	63.6	69.1
DLRNN Duan, He, and Zhao (2017)	77.2	64.9	70.5
ANN-S2 S. Liu, Chen, Liu, and Zhao (2017)	78.0	66.3	71.7
GMLATT J. Liu, Chen, Liu, and Zhao (2018)	78.9	66.9	72.4
GCN-ED T. H. Nguyen and Grishman (2018)	77.9	68.8	73.1
Lu's DISTILL Lu, Lin, Han, and Sun (2019)	76.3	71.9	74.0
TS-DISTILL J. Liu, Chen, and Liu (2019)	76.8	72.9	74.8
DMBERT* X. Wang, Han, et al. (2019b)	77.6	71.8	74.6
AD-DMBERT* X. Wang, Han, et al. (2019b)	77.9	72.5	75.1
DRMM* Tong, Wang, et al. (2020)	77.9	74.8	76.3
GatedGCN* Lai, Nguyen, and Nguyen (2020b)	78.8	76.3	77.6
EKD* Tong, Xu, et al. (2020)	79.1	78.0	78.6
GPTEDOT*	82.3	76.3	**79.2**

Table 24. Performance on the on ACE 2005 test set. * indicates models that use BERT for the encoding.

For each dataset, we use its training and development data to fine-tune the GPT-2 model. We tune the hyperparameters for the proposed teacher-student architecture using a random search. All the hyperparameters are selected based on the F1 scores on the development set of the ACE 2005 dataset. The same hyperparameters from this fine-tuning are then applied for other datasets for consistency. In our model we use the small version of GPT-2 to generate data. In the base model, we use BERT$_{base}$, 300 dimensions in the hidden states of BiLSTM and 2 layers of feed-forward neural networks with 200 hidden dimensions to predict events. The trade-off parameters τ, α, β and γ are set to 0.1, 0.1, 0.05, and 0.08, respectively. The learning rate is set to 0.3 for the Adam optimizer and the batch size of 50 are employed during training. Finally, note that we do not update the BERT model for word embeddings in this work due to its better performance on the development data of ACE 2005.

Model	P	R	F1
CNN T. H. Nguyen and Grishman (2015a)	51.8	36.7	43.0
DMCNN Y. Chen et al. (2015)	47.5	38.7	43.2
GCN-ED T. H. Nguyen and Grishman (2018)	46.3	51.8	48.9
MOGANED Yan, Jin, Meng, Guo, and Cheng (2019b)	53.7	59.6	56.5
CyberLSTM Satyapanich, Ferraro, and Finin (2020b)	42.5	29.0	34.5
DMBERT X. Wang, Han, et al. (2019b)	59.4	51.3	55.1
BERT-ED Man Duc Trong et al. (2020)	60.2	56.1	58.1
GPTEDOT	65.9	64.1	**65.0**

Table 25. Comparison with state-of-the-art models on CySecED. All the models in this table use BERT.

Model	P	R	F1
DMBERT X. Wang, Han, et al. (2019b)	62.6	44.0	51.7
GatedGCN Lai, Nguyen, and Nguyen (2020b)	66.5	59.0	62.5
GPTEDOT	55.5	78.6	**65.1**

Table 26. Model's performance on RAMS. All the models use BERT in this table.

4.2.2 Results. Results of experiments on the ACE 2005 test set are shown in Table 24. The most important observation is that the proposed model GPTEDOT significantly outperforms all the baseline models ($p < 0.01$), thus showing the benefits of GPT-generated data and the teacher-student framework with knowledge consistency for ED in this work. In particular, compared to the BERT-based models that leverage data augmentation, i.e., AD-DMBERT X. Wang, Han, et al. (2019b) with semi-supervised and adversarial learning, DRMM Tong, Wang, et al. (2020) with image-enhanced models, and EKD Tong, Xu, et al. (2020) with external open-domain event triggers, the better performance of GPTEDOT highlights the advantages of GPT-2 to generate data for ED models.

Results of experiments on the CySecED test set are presented in Table 25. This table reveals that the teacher-student architecture GPTEDOT significantly improves the performance over previous state-of-the-art models for ED in

cybersecurity domain. This is important as it shows that the proposed model is effective in different domains. In addition, our results also suggest that GPT-2 can be employed to generate effective data for ED in domains where data annotation for ED requires extensive domain expertise and expensive cost to obtain such as the cybersecurity events. Moreover, the higher margin of improvement for GPTEDOT on CySecED compared to the those on the ACE 2005 dataset suggests the necessity of using more training data for ED in technical domains.

Finally, results of experiments on the RAMS test set are reported in Table 26. Consistent with our experiments on ACE 2005 and CySecED, our proposed model achieve significantly higher performance than existing state-of-the-art models ($p < 0.01$), thus further confirming the advantages of GPTEDOT for ED.

4.2.3 Ablation Study. This ablation study evaluates the effectiveness of different components in GPTEDOT for ED. First, for the importance of the generated data $\mathcal{G}$ from GPT-2 and the teacher-student architecture to mitigate noises, we examine the following baselines: (1) $\textbf{Base}^{\mathcal{O}}$: The baseline is the base model trained only on the original data $\mathcal{O}$, thus being equivalent to the teacher model and not using the student model; and (2) $\textbf{Base}^{\mathcal{A}}$: This baseline trains the base model on the combination of the original and generated data, i.e., $\mathcal{A}$, using the multi-learning setting (i.e., the teacher model is excluded).

Second, for the multi-task learning design in the teacher network, we explore the following ablated models: (3) $\textbf{Teacher}^{-A}$: This baseline removes the auxiliary task EI in the teacher from GPTEDOT. As such, the OT-based knowledge consistency for EI is also eliminated; (4) $\textbf{Teacher}^{-M}$: In this model, the main task ED is utilize to train the teacher, so the corresponding KL-based knowledge consistency for ED is also removed.

Third, for the design of the knowledge consistency losses in the student network, we evaluate the following baselines: (5) **Student**$^{-OT}$: This ablated model eliminates the OT-based knowledge consistency loss for the auxiliary task EI in the student's training of GPTEDOT (the auxiliary task is still employed for the teacher and the student); (6) **Student**$^{-KL}$: For this model, the KL-based knowledge consistency for the main task ED is ignored in the student's training; (7) **Student**$^{+OT}$: In this baseline, we use OT for the knowledge consistency on both the main and the auxiliary tasks. Here, for the main task ED, the cost function $C(x, y)$ for OT is still obtained via the Euclidean distances between representation vectors while the distributions $p(x)$ and $p(y)$ are based on the maximum probabilities of the label-probability distributions $P_s(.|X_o, t_o)$ and $P_s(Y_g, t_g)$ for the ED task; and (8) **Student**$^{+KL}$: This baseline employs the KL divergence between models' predicted distributions to enforce the teacher-student consistency for both the main task and the auxiliary task. To this end, for the auxiliary task EI, we convert the final activation of $\texttt{FF}_{aux}$ into a distribution with two data points (i.e., $[\texttt{FF}_{aux}(X), 1 - \texttt{FF}_{aux}(X)]$) to compute the KL divergence between the teacher and the student.

Finally, for the importance of Euclidean distances and event trigger likelihoods in the OT-based distance between $\mathcal{O}$ and $\mathcal{G}$ for knowledge consistency in EI, we investigate two baselines: (9) **OT**$^{-Rep}$: Here, to compute OT, we use constant cost between every pair of sentences, i.e., $C(x, y) = 1$ (i.e., ignoring representation-based distances); and (10) **OT**$^{-Score}$: This model uses uniform distributions for $p(x)$ and $q(y)$ to compute the OT (i.e., ignoring event trigger likelihoods).

Model	P	R	F1
GPTEDOT (full)	82.4	75.0	**78.5**
Base$^{\mathcal{O}}$	78.2	73.7	75.9
Base$^{\mathcal{A}}$	75.8	73.9	74.9
Teacher^{-A}	76.9	78.1	77.5
Teacher^{-M}	75.8	77.9	76.9
Student^{-}OT	75.4	79.3	77.3
Student^{-}KL	76.8	77.3	77.0
Student^{+OT}	76.1	76.6	76.4
Student^{+KL}	77.1	76.7	76.9
OT^{-Rep}	76.8	77.3	77.0
OT^{-Score}	78.0	77.1	77.6

Table 27. Ablation study on the ACE 2005 dev set.

Dataset	Sentence
ACE 2005	I was totally shocked by the court's decision to agree with Sam Sloan after he TRG$_s$ **sued** TRG$_e$ his children.
CySecED	According to the last update by the company, the following techniques are used to protect against such TRG$_s$ **malware** TRG$_e$.
RAMS	The Russian officials TRG$_s$ **vowed** TRG$_e$ to bomb the ISIS bases after the last week's TRG$_s$ **attack** TRG$_e$.

Table 28. Generated sentences by GPT-2 for different datasets. Event triggers are shown in boldface that are surrounded by the special tokens TRG$_s$ and TRG$_e$ generated by GPT-2.

Error Type	Sentence Example	Proportion
Incompleteness	A federal judge on Monday settled TRG$_s$ **charges** TRG$_e$ against seven members of	18%
Repetition	Do you think the TRG$_s$ **attack** TRG$_e$ will happen to you or do you think the TRG$_s$ **attack** TRG$_e$ will happen to you?	15%
Inconsistency	this morning we were watching the news and heard the news about the tragic TRG$_s$ **death** TRG$_e$ of a young boy and her mother in Iraq.	12%
Missing Labels	Aaron Tramailer's story is the story of a woman who was forced into **suicide**.	29%
Incorrect Labels	The SEC is a very good place to TRG$_s$ hide TRG$_e$ money.	26%

Table 29. Samples of noisy generated sentences for the ACE 2005 dataset from GPT-2. Event triggers are shown in boldface and the special tokens TRG$_s$ and TRG$_e$ are generated by GPT-2.

$\lvert\mathcal{G}\rvert$	P	R	F1
0.5 * $\lvert\mathcal{O}\rvert$	80.3	72.4	76.2
1.0 * $\lvert\mathcal{O}\rvert$	82.4	75.0	78.5
2.0 * $\lvert\mathcal{O}\rvert$	81.3	73.3	77.1
3.0 * $\lvert\mathcal{O}\rvert$	78.4	71.8	75.0

Table 30. The performance of GPTEDOT on the ACE 2005 dev set with different sizes of the generated data $\mathcal{G}$.

We report the performance of the models (on the ACE 2005 development set) for the ablation study in Table 27. There are several observations from this table. First, the generated data $\mathcal{G}$ and the teacher-student architecture are necessary for GPTEDOT to achieve the highest performance. In particular, comparing with Base$^{\mathcal{O}}$, the better performance of GPTEDOT indicates the benefits of the GPT-generated data. Moreover, the better performance of Base$^{\mathcal{O}}$ over Base$^{\mathcal{A}}$ reveals that the simple combination of the synthetic and original data without any effective method to mitigate noises might be harmful. Second, the lower performance of Teacher^{-A} and Teacher^{-M} shows that both the auxiliary and the main task (i.e., multi-task learning) in the teacher are integral to produce the best performance. Third, the choice of methods to promote knowledge consistency is important and the proposed combination of KL and OT for the ED and EI tasks (respectively) are necessary. In particular, removing or replacing each of them with the other one (i.e., Student^{+OT} and Student^{+KL}) would decrease the performance significantly. Finally, in the proposed consistency method based on OT for EI, it is beneficial to employ both representation-level distances (i.e., OT^{-Rep}) and models' predictions for event trigger likelihoods (i.e., OT^{-Score}) as removing any of them hurts the performance.

4.2.4 Analysis. To provide more insights into the quality of the synthetic data $\mathcal{G}$, we provide samples of sentences that are generated by the fine-tuned GPT-2 model on each dataset in Table 28. This table illustrates that the generated sentences also belong to the domains of the original data (i.e., the cybersecurity domain). As such, combining synthetic data with original data is promising for improving ED performance as demonstrated in our experiments.

As discussed earlier, the generated data $\mathcal{G}$ is not free of noise. In order to better understand the types of errors existing in generated sentences, we manually assess 200 sentences randomly selected from the set $\mathcal{G}$ generated by the fine-tuned GPT-2 model on the ACE 2005 dataset. We categorize the errors into five types and provide their proportions along with example for each error type in Table 29. This table shows that the majority of errors are due to missing labels (i.e., no special tokens TRG_s and TRG_e are generated) or incorrect labels (i.e., marked words are not event triggers of interested types) generated by the language model.

Finally, to study the importance of the size of the generated data to augment training set for ED, we conduct an experiment in which different numbers of generated samples in $\mathcal{G}$ (for the ACE 2005 dataset) are combined with the original data $\mathcal{O}$. The results are shown in Table 30. According to this table, the highest performance of the proposed model is achieved when the numbers of the generated and original data are equal. More specifically, decreasing the number of generated samples potentially limits the benefits of data augmentation. On the other hand, increasing the size of generated data might introduces extensive noises and become harmful to the ED models.

4.3 Related Work

Early methods for ED have employed feature-based techniques (Ahn, 2006; Hong et al., 2011a; Ji & Grishman, 2008; Q. Li et al., 2013; Liao & Grishman, 2010a, 2010b; McClosky et al., 2011; Miwa et al., 2014; Patwardhan & Riloff, 2009; B. Yang & Mitchell, 2016a). Later, advanced deep learning methods (Y. Chen et al., 2015; T. H. Nguyen et al., 2016a; T. H. Nguyen & Grishman, 2015a; T. H. Nguyen, Sil, Dinu, & Florian, 2016b; T. M. Nguyen & Nguyen, 2019a; Sha et al., 2018; S. Yang et al., 2019; J. Zhang et al., 2019; Y. Zhang, Xu, et al., 2020)

have been applied for ED. One challenge for ED research is the limited size of existing datasets that hinder the training of effective models. Prior works have attempted to address this issue via unsupervised L. Huang et al. (2016); Yuan et al. (2018), semi-supervised Ferguson, Lockard, Weld, and Hajishirzi (2018); R. Huang and Riloff (2012); Liao and Grishman (2010a), distantly supervised Araki and Mitamura (2018); Keith et al. (2017); M. Nguyen and Nguyen (2018b); Y. Zeng et al. (2017), and few/zero-shot B. Huang, Ou, and Carley (2018); Lai, Dernoncourt, and Nguyen (2020a, 2020b) learning. In this work, we propose a novel method to augment training data for ED by exploiting the powerful language model GPT-2 to automatically generate new samples.

Leveraging GPT-2 for augmenting training data has also been studied for other NLP tasks recently (e.g., relation extraction, commonsense reasoning) Anaby-Tavor et al. (2020); Bosselut et al. (2019); Kumar, Choudhary, and Cho (2020); Madaan et al. (2020); Papanikolaou and Pierleoni (2020); B. Peng, Zhu, Zeng, and Gao (2020); Y. Yang et al. (2020); D. Zhang, Li, Zhang, and Yin (2020). However, none of those works has explored GPT-2 for ED. In addition, existing methods only resort to heuristics to filter out noisy samples generated by GPT-2. In contrast, we propose a novel differentiable method capable of preventing noises from diverging representation vectors of the models for ED.

4.4 Conclusion

We propose a novel method for augmenting training data for ED using the samples generated by the language model GPT-2. To avoid noises in the generated data, we propose a novel teacher-student architecture in a multi-task learning framework. We introduce a mechanism for knowledge consistency enforcement to mitigate noises from generated data based on optimal transport. This mechanism

is based on the inferring structure between generated and original labeled data. Experiments on various ED benchmark datasets demonstrate the effectiveness of the proposed method. Finally, in the last Chapter, we will employ the findings of the methods presented in the previous Chapters to improve IE in a less-explored setting, i.e., Cross-Lingual Information Extraction.

CHAPTER V

EXPLORING STRUCTURES FOR IE IN CROSS-LINGUAL TRANSFER LEARNING

This Chapter contains materials from the published paper: "*Amir Pouran Ben Veyseh, Javid Ebrahimi, Franck Dernoncourt, and Thien Nguyen. "**MEE: A Novel Multilingual Event Extraction Dataset'**. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, 9603–13. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, 2022. https://aclanthology.org/2022.emnlp-main.652*". In this publication, the experiments were entirely done by the author of this dissertation, Amir Pouran Ben Veyseh. The other co-authors provided feedback regarding the experiments and results. Amir wrote the entire paper and Dr. Thien Huu Nguyen provided editorial feedback for this paper.

In the final Chapter, we provide a case study on how structural information could be helpful to improve the performance of IE models in less-explored settings. Specifically, cross-lingual event extraction (EE) is an important task for IE that aims to exploit the labeled data in other languages to train an IE model for a target language with less annotated data. Since there is no labeled data that covers a diverse set of languages with enough labeled data, we first discuss the details of collecting and annotating a new dataset for event argument extraction in multiple languages. Next, in the experiments, we show that structure-based models would outperform the sequential models for cross-lingual EE.

Event Extraction (EE) is one of the major tasks of Information Extraction (IE) for text. In a complete EE pipeline, three major goals should be pursued: (1) Entity Mention Detection (EMD): to recognize mentions of real world entities; (2)

Event Detection (ED): to identify event mentions/triggers and their types. An event trigger is a word or phrase that most clearly refers to the occurrence of an event; and (3) Event Argument Extraction (EAE): to find participants/arguments of an event mentioned in text. A participant is an entity mention that has an specific role in a given event mention. For instance, in the sentence "*The soldiers were hit by the forces.*", there are two entity mentions "*soldiers*" and "*forces*" of types *PERSON* and *ORGANIZATION* and an event trigger "*hit*" of type *ATTACK*. Also, the two event mentions "*soldiers*" and "*forces*" play the argument roles of *Victim* and *Attacker* (respectively) in the *ATTACK* event. An EE system could be employed in other downstream applications such as Question Answering, Knowledge Base Population and Text Summarization to assist extracting information about events in text.

Multiple methods have been proposed for Event Extraction. Early work has employed feature-based models Ahn (2006); Hong et al. (2011b); Ji and Grishman (2008); Q. Li et al. (2013); Liao and Grishman (2010a); B. Yang and Mitchell (2016b) while later methods have explored deep learning to present state-of-the-art performance for Event Extraction Y. Chen et al. (2015); Lai, Nguyen, and Nguyen (2020b); Y. Lin, Ji, Huang, and Wu (2020); M. V. Nguyen, Lai, and Nguyen (2021b); T. H. Nguyen, Cho, and Grishman (2016); T. H. Nguyen and Grishman (2015b); Sha et al. (2018); X. Wang, Han, et al. (2019b). However, despite all advancements on event extraction in recent years, a major limitation of current EE research is to overly focus on a few popular languages, thus failing to adequately reveal challenges and generalization of models in many other languages of the world. As such, a critical barrier for studying EE over multiple languages is the lack of high quality datasets that fully annotate data for many other languages for EE. For

instance, the most popular dataset for EE, i.e., ACE 2005 Walker et al. (2006a), only provide annotations for three languages English, Chinese and Arabic while TAC KBP datasets Mitamura, Liu, and Hovy (2016, 2017) only supports English, Chinese and Spanish. The TempEval-2 dataset Verhagen, Saurí, Caselli, and Pustejovsky (2010) involves 6 languages; however, it does not offer event argument annotation. Even worse, recently created datasets, e.g., MAVEN X. Wang et al. (2020), RAMS Ebner et al. (2020), and WikiEvents, are only annotated for English. In all, such language and task limitations prevents research to comprehensively develop and evaluate EE methods over different languages and multilingual settings. Moreover, the limited size of these datasets, i.e. less than 11K and 27K in ACE 2005 and TempEval-2 respectively, hinders training of data-hungry deep learning models. Finally, we note that important multilingual datasets for EE, e.g., ACE 2005 and TAC KBP, are not publicly available, which further restricts research on this domain.

To address these limitations, in this work, we propose a large-scale Multilingual Event Extraction (MEE) dataset that covers 8 typologically different languages from multiple language families, including English, Spanish, Portuguese, Polish, Turkish, Hindi, Korean, and Japanese. As such, Portuguese, Polish, Turkish, Hindi, and Japanese are not explored in the popular multilingual datasets for EE, i.e., ACE 2005 and TAC KBP. Importantly, to enable public data sharing and diversity the data, we employ Wikipedia articles for the 8 languages in diverse topics (i.e., Economy, Politics, Technology, Crime, Nature and Military) for EE annotation.

Our dataset comprehensively annotates each document in a language for all the three sub-tasks EMD, ED, and EAE. To be consistent with prior EE

Category	English	Portuguese	Spanish	Polish	Turkish	Hindi	Japanese	Korean
Economy	1,095	112	168	315	297	189	199	250
Politics	3,202	308	772	1,270	1,233	349	232	248
Technology	2,171	189	400	712	815	295	312	249
Crimes	893	78	220	152	118	95	80	73
Nature	1,195	398	705	455	398	245	299	185
Military	4,444	415	1,003	1,575	1,619	326	378	495
Total	13,000	1,500	3,268	4,479	4,480	1,499	1,500	1,500

Table 31. Numbers of annotated segments in each Wikipedia subcategory for our 8 languages.

research, we inherit the type anthologies for such tasks from the ACE 2005 dataset that provides well-designed guidelines and examples for the types. In particular, we include 7 entity types, 8 event types and 16 event sub-types, along with 23 argument roles in MEE to facilitate EE annotation over multiple languages. Overall, our dataset involves more than 415K entity mentions, 50K event triggers, and 38K arguments, which are much larger than previous multilingual EE datasets to better support model training and evaluation with deep learning.

Due to shared information schema over all the languages, our MEE dataset enables cross-lingual transfer learning evaluation of MEE models where training and test data comes from different languages. To this end, we conduct comprehensive experiments for both monolingual and cross-lingual learning settings to provide insights for language-specific challenges and cross-lingual generalization of EE methods. By examining both pipeline and joint inference models for EE, our experiments show that the proposed dataset present unique challenges with less satisfactory performance of existing EE models, especially for cross-lingual settings, thus calling for more research efforts for multilingual EE in the future.

5.1 Data Annotation

We follow the entity/event type definition and annotation guidelines from the popular ACE 2005 dataset to benefit from its well-designed documentation and

be consistent with prior EE research. As such, entity mentions refer to mentions of real-world entities in text that can be expressed via names, nominals, and pronouns. Entity Mention Extraction (EMD) is more general than Named Entity Recognition that only concerns names of entities. In addition, an event is defined as an incident whose occurrence changes the state of real world entities. An event mention is the part of input text that refers to an event that consists of two components: (1) Event Trigger: the words that most clearly refer to the occurrence of the event. It is noteworthy that we allow an event trigger to span multiple words to accommodate trigger annotation for multiple languages. For instance, in the Turkish phrase "*tayin etmek*", both words are necessary to indicate an event trigger of type "*Appoint*"; and (2) Event Arguments: the entity mentions that are involved in the event with some roles.

Based on the ACE 2005 dataset, our dataset annotates entity mentions for 7 entity types: **PERSON** (human entities), **ORGANIZATION** (corporations, agencies, and other groups of people), **GPE** (geographical regions defined by political and/or social groups), **LOCATION** (geographical entities such as landmasses or bodies of water), **FACILITY** (buildings and other permanent man-made structures), **VEHICLE** (physical devices primarily designed to move an object from one location to another), and **WEAPON** (physical devices primarily used as instruments for physically harming). For event types, to avoid confusion and improve data quality, we prune the original ACE 2005 event types to only include the types that are not ambiguous across multiple languages. For instance, in Turkish, the event types *Sentence* and *Convict* are very similar (both can be evoked by the phrase "*Mahkum etmek*") so they are not retained in our dataset. As such, we preserve 8 event types and 16 sub-types that are distinct enough

for annotation in our dataset. Finally, for event arguments, we preserve all 23 argument roles in the ACE 2005 dataset. Table 34 shows the list of event types along with their argument roles in our dataset.

5.1.1 Data Preparation. Our dataset MEE covers 8 different languages, i.e., English, Spanish, Portuguese, Polish, Turkish, Hindi, Korean and Japanese. These languages are selected based on their diversity in terms of typology and their novelty with respect to existing multilingual EE datasets. For each language, we employ its latest dump of Wikipedia articles as raw data for annotation. To focus on event data, we select articles in the sub-categories under category *Event* in Wikipedia. In particular, the following sub-categories are considered to improve topic diversity: Economy, Politics, Technology, Crimes, Nature, and Military. Note that we start with these categories in English Wikipedia. Afterward, we follow interlinks between the categories in different languages to locate the intended categories for Wikipedia for non-English languages in MEE.

We process the collected articles with the WikiExtractor tool Attardi (2015) to obtain clean textual data and meta-data for each article. The textual data is then split into sentences and tokenized into words by the multilingual NLP toolkit Trankit M. V. Nguyen, Lai, Veyseh, and Nguyen (2021). Afterward, to annotate the data with entity and event mentions, one approach is to directly ask annotators to read each article entirely for annotation. However, as the articles in Wikipedia might be lengthy, this approach can be overwhelming for annotators, thus hindering their attention and lowering quality of annotated data. To address this issue, we follow prior dataset creation efforts for EE, i.e., RAMS Ebner et al. (2020), to divide the articles into segments of five consecutive sentences. Each segment will

Language	#Seg.	Avg. Length	#Entities	#Triggers	#Arguments	Challenging Entity Type	Challenging Trigger Type	Language Family
English	13,000	123	190,592	17,642	13,548	GPE	Personnel	Germanic
Spanish	3,268	112	48,001	6,064	802	GPE	Conflict	Italic
Portuguese	1,500	102	25,463	1,953	12,329	Location	Personnel	Italic
Polish	4,479	108	62,971	10,875	3,395	Facility	Transaction	Balto-Slavic
Turkish	4,480	117	38,469	8,390	1,416	GPE	Personnel	Turkic
Hindi	1,499	98	18,797	1,810	2,117	Facility	Conflict	Indo-Iranian
Japanese	1,500	99	19,174	2,152	3,399	Location	Personnel	Japonic
Korean	1,500	103	12,508	1,125	1,742	GPE	Personnel	Koreanic
Total (MEE)	31,226	-	415,975	50,011	38,748	-	-	-

Table 32. Statistics of the MEE dataset. #Seg. represents the numbers of annotated text segments for each language. All annotated segments consist of 5 sentences and their lengths (Avg. Length) are computed in terms of numbers of tokens. "*Challenging Type*" indicates the types where entity or event trigger annotation involves largest disagreement between annotators in each language.

then be annotated separately for EE tasks so annotators can better capture the entire context to provide entity and event annotation. Note that similar to RAMS, we annotate all event arguments in a text segment for each event trigger, thus allowing event arguments to appear in different sentences from the event trigger (i.e., document-level EAE). Finally, to accomodate our budget, a sample of text segments is obtained for each language for annotation. The numbers of selected text segments for each category per language in our dataset are presented in Table 31.

5.1.2 Annotation Process. To annotate the sampled article segments, we employ the crowd-sourcing platform `upwork.com` that allows us to hire freelancers across the globe with different expertise. For each language in our dataset, we choose native speakers as annotator candidates. In addition, we require them to be fluent in English, have experience in related tasks (i.e., data annotation for information extraction), and have approval rate higher than 95% (i.e., provided in their profiles). The candidates are first provided with annotation guidelines and interfaces in English. Afterward, they are invited to an annotation test for entity mentions, event triggers, and arguments. Those candidates who correctly annotate all test cases are then officially hired to work on our annotation

Language	#Annotator	EMD	ED	EAE
English	10	0.792	0.834	0.820
Spanish	10	0.788	0.812	0.823
Portuguese	5	0.791	0.803	0.799
Polish	8	0.780	0.799	0.813
Turkish	10	0.785	0.813	0.822
Hindi	6	0.790	0.803	0.812
Japanese	5	0.793	0.789	0.780
Korean	6	0.802	0.810	0.825

Table 33. Number of annotators and agreement scores for 8 languages in MEE for Entity Mention Detection (EMD), Event Detection (ED) and Event Argument Extraction (EAE).

jobs. Table 33 shows the numbers of annotators who are hired to annotate data for each language in our dataset. Next, before the actual annotation process, the English annotation guideline and examples are translated to each target language by the hired annotators. Any language-specific confusions and rules for annotation is discussed and included in the translation to create a common understanding. Finally, our language experts will review the annotation guideline in each language to avoid conflicts across languages to be used for actual annotation.

Our annotation process is done in three separate steps to annotate data for three EE tasks with entity mentions, event triggers, and event arguments in this order. In particular, the annotation for a later task will be performed over the text segments that have been annotated and finalized for previous tasks (e.g., event arguments will be annotated over segments that are already provided with entity mentions and event triggers). As such, for each task, 20% of text segments for each language will be co-annotated by the annotators to measure agreement score. The remaining 80% of text segments will be distributed and annotated separately by the annotators for each language. Based on the Krippendorff's alpha Krippendorff (2011) with MASI distance metric Passonneau (2006), we report the inter-annotator

ID	Event	Arguments
1	Life_Be-Born	Person, Time, Place
2	Life_Marry	Person, Time, Place
3	Life_Divorce	Person, Time, Place
4	Life_Injure	Agent, Victim, Instrument, Time, Place
5	Life_Die	Agent, Victim, Instrument, Time, Place
6	Movement_Transport	Agent, Artifact, Vehicle, Price, Origin, Destination, Time
7	Transaction_Transfer-Ownership	Buyer, Seller, Beneficiary, Price, Artifact, Time, Place
8	Transaction_Transfer-Money	Giver, Recipient, Beneficiary, Money, Time, Place
9	Business_Start-Organization	Agent, Organization, Time, Place
10	Conflict_Attack	Attacker, Target, Instrument, Time, Place
11	Conflict_Attack	Entity, Time, Place
12	Contact_Meet	Entity, Time, Place
13	Contact_Phone-Write	Entity, Time
14	Personnel_Start-Position	Person, Entity, Position, Time, Place
15	Personnel_End-Position	Person, Entity, Position, Time, Place
16	Justice_Arrest-Jail	Person, Agent, Crime, Time, Place

Table 34. Event types and argument roles for each type in MEE. The types and roles are inherited from the event extraction annotation guideline in the ACE 2005 dataset Walker et al. (2006a).

agreements (IAA) for each task and language in Table 33, showing high agreement scores and quality of our MEE dataset. Note that after independent annotation for each EE task, the annotators also share their annotations and communicate with each other to resolve any conflicts and finalize our data.

5.1.3 Data Analysis. Table 32 shows the main statistics of MEE for each language. As such, comparing to the popular multilingual ACE 2005 dataset Walker et al. (2006a) for EE, our MEE dataset provides more languages (i.e., 3 vs. 8) and much more event mentions (i.e., 11K vs. 50K). For other multilingual datasets for EE, i.e., TAC KBP (with three languages and 6.5K event mentions) Mitamura et al. (2016, 2017) and TempEval-2 (with 6 languages and 27K event mentions) Verhagen et al. (2010), they do not annotate entity mentions and event arguments. In contrast, our MEE dataset fully annotates texts for three EE tasks (i.e., EMD, ED, and EAE) and also with more languages and event mentions.

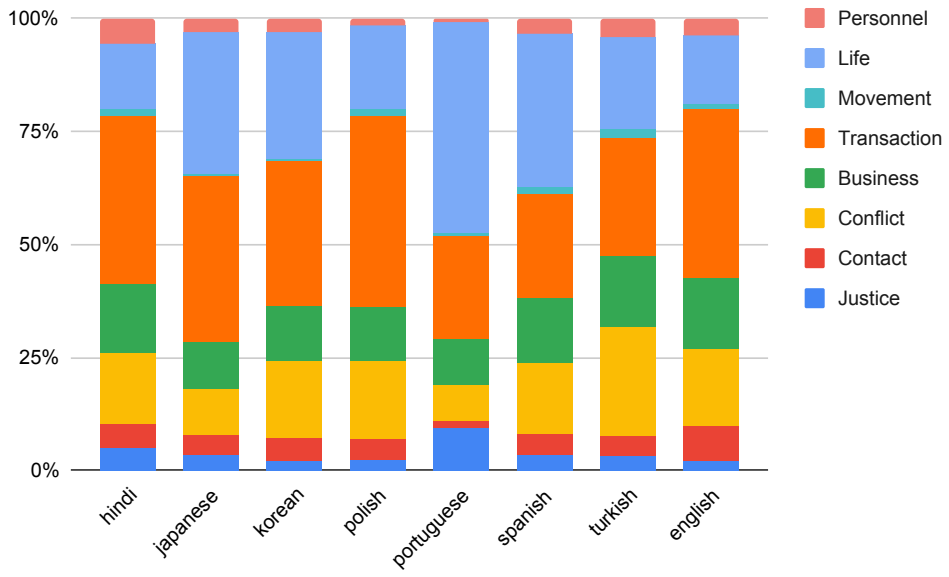

Figure 3. Distributions of event types in each language.

This clearly demonstrates the advantages of our dataset over existing multilingual datasets for EE.

In addition, from the table, we find that the languages in our dataset exhibits diverse densities for entity mentions, event triggers, and arguments in texts. In particular, while the average number of entities in a text segment in Portuguese is 16.9, this number is only 8.3 in Korean. For event density, in Polish, there are 2.4 event mentions per article segment on average while the average number in Korean is only 0.75. Similarly for event arguments, the average number of arguments per event is 6.1 in Portuguese and only 0.75 in English. Further, Table 32 highlights the divergences between languages regarding challenging entity and event types. Specifically, we employ the disagreement rates (i.e., number of disagreements divided by frequency of mentions) between annotators for each entity and event types. Those types that have highest disagreement rates are selected as challenging entity or event types. Finally, Figures 4, 3, and 5 present the distributions of entity

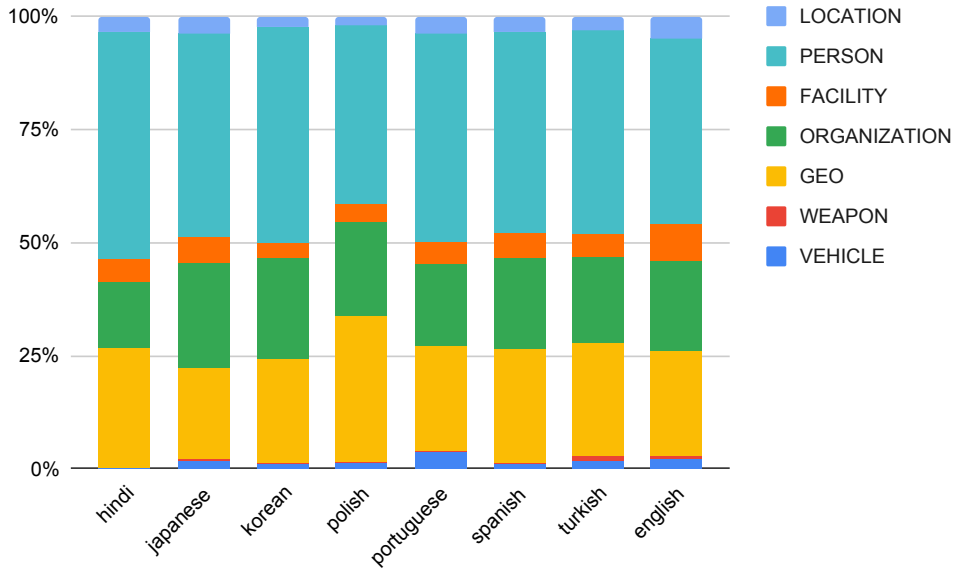

Figure 4. Distributions of entity types in each language.

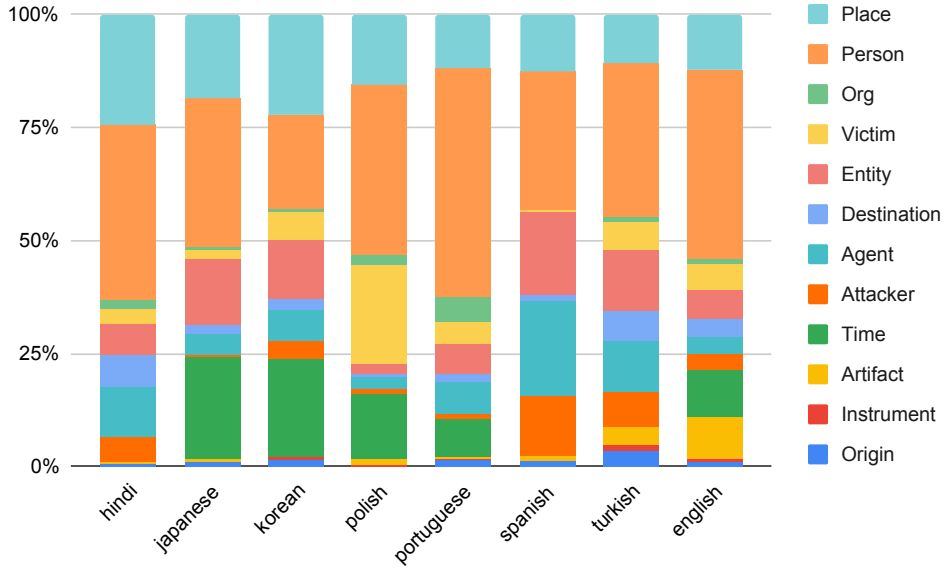

Figure 5. Distributions of most argument roles for each language.

Language	Pipeline			OneIE			FourIE		
	Entity	Event	Argument	Entity	Event	Argument	Entity	Event	Argument
English	70.32	70.58	61.14	62.18	70.09	62.94	69.72	72.19	65.89
Spanish	70.39	66.19	60.16	70.27	65.00	60.31	71.89	67.49	62.19
Portuguese	75.13	71.33	69.15	73.19	70.13	71.27	74.98	72.99	70.17
Polish	69.27	59.12	60.09	60.09	59.44	60.14	68.23	60.98	61.32
Turkish	71.88	66.09	56.19	71.98	61.27	58.72	72.33	65.13	59.80
Hindi	66.22	57.77	57.78	61.72	58.18	59.44	65.23	59.88	60.82
Japanese	68.19	67.89	68.19	71.40	65.01	63.17	70.88	66.88	70.19
Korean	57.17	61.26	67.87	55.87	61.10	65.41	58.18	60.09	69.23
Avg.	68.57	65.03	62.57	65.84	63.78	62.68	68.93	65.70	64.95

Table 35. Performance (F1 scores) of models in the monolingual setting using mBERT on MEE.

types, event types, and argument roles (respectively) for each language in our dataset, which further demonstrate the differences between languages in MEE. In all, such differences over various dimensions will cause significant challenges for EE models to adapt to new languages (e.g., for cross-lingual transfer learning), thus presenting ample opportunities for multilingual EE research with our dataset.

5.2 Experiments

This section evaluates the state-of-the-art models for Event Extraction to reveal challenges in our new dataset MEE. To this end, the annotated article segments for each language in MEE are randomly split into training/development/test portions with the ratios of 80/10/10. Here, to prevent any information leakage, we ensure that different segments of an article (if any) are only assigned to one portion of the data split for each language. We examine EE models in two different settings: (1) monolingual learning where training and test data of models comes from the same language; (2) cross-lingual transfer learning where models are trained on training data of one language (i.e., the source language), but evaluated directly on test data of the other languages (i.e., the target languages).

Language	Pipeline			OneIE			FourIE		
	Entity	Event	Argument	Entity	Event	Argument	Entity	Event	Argument
English	70.22	71.28	66.34	70.39	70.29	68.68	71.19	73.14	68.23
Spanish	70.33	64.32	61.12	70.18	62.46	62.23	72.87	65.90	63.11
Portuguese	70.39	71.88	71.75	72.16	69.43	70.33	73.98	70.43	72.23
Polish	69.14	60.45	61.23	72.22	63.77	60.15	70.25	62.87	62.84
Turkish	76.13	67.18	55.78	74.45	65.31	57.40	75.19	67.29	58.23
Hindi	65.14	59.34	58.22	61.72	58.18	59.44	66.69	61.99	62.19
Japanese	71.34	67.77	69.19	68.20	62.89	70.90	72.82	65.27	73.55
Korean	59.13	62.34	69.70	59.99	60.55	66.89	60.24	61.18	70.09
Avg.	68.98	65.57	64.17	68.84	64.36	64.57	70.40	66.01	66.31

Table 36. Performance (F1 scores) of models in the monolingual setting using XLM-RoBERTa on MEE.

Models: We evaluate two typical approaches for EE models with pipeline and joint inference in this work. First, for the pipeline approach, a model is trained separately for each of the three tasks in EE, i.e., entity mention detection (EMD), event detection (ED), and event argument extraction (EAE). Here, the EMD and ED tasks are modeled as sequence labeling problems, aiming to predict BIO tag sequences for each input sentence to capture spans and types of entity and event mentions. As such, motivated by previous work X. Wang et al. (2020), our EMD and ED models leverage a pre-trained transformer-based language model to encode the input text. The representation for each token in input text (obtained via average of hidden vectors of word-pieces in the last transformer layer) is then sent into a feed-forward network to compute a tag distribution for the token for training and decoding. For EAE, the task is formulated as a text classification problem in which the input consists of an input text and two word indices for the positions of an event trigger and an entity mention of interest. The goal is to predict the argument role that the entity mention plays for the event. To this end, we also use a pre-trained language model to obtain representations for the tokens in input text. Next, the representations for the event trigger and entity mention words are concatenated and sent to a feed-forward network to predict argument role. Note

that the EAE model employs golden entity mentions and event triggers during the training process while the outputs from the EMD and ED models are fed into the EAE model in the test time.

Second, for the joint inference approach, EE models simultaneously predicts entity mentions, event triggers, and arguments in end-to-end fashion to avoid error propagation and leverage inter-dependencies between tasks. To this end, we evaluate two state-of-the-art (SOTA) joint EE models, OneIE Y. Lin et al. (2020) and FourIE M. V. Nguyen, Lai, and Nguyen (2021b), in this work due to their language-agnostic nature. Both OneIE and FourIE utilize pre-trained language models to represent input texts and capture cross-task dependencies for joint inference. However, FourIE employs structural information, i.e. dependency tree, to encode the input text. Note that these models are original designed to include the relation extraction task between entities. To adapt them to EE, we obtain their implementations from the original papers and remove the relation extraction components. Finally, for performance measure, we report the performance (F1 scores) of EE models over three tasks EMD (Entity), ED (Event), and EAE (Argument) using the same correctness criteria as in prior work Y. Lin et al. (2020) (i.e., requiring correct prediction for both offsets and types of entity mentions, event triggers, and argument roles).

Hyper-parameters: To facilitate evaluation with multiple languages, we leverage the multilingual pre-trained language models (PLMs) mBERT Devlin, Chang, Lee, and Toutanova (2019b) and XLM-RoBERTa Conneau et al. (2020) (base versions) to encode texts for EE models. For the pipeline approach, we fine-tune the hyper-parameters for the EMD, ED, and EAE models over development data for English and apply the selected values for all experiments for consistency. In particular, our

Language	XLM-RoBERTa			mBERT		
	Entity	Event	Argument	Entity	Event	Argument
English	69.72	72.19	65.89	71.19	73.14	68.23
Spanish	61.96	59.70	52.23	60.72	60.06	50.77
Portuguese	59.98	54.80	52.23	56.17	52.98	50.28
Polish	52.89	51.78	52.44	53.44	50.29	53.56
Turkish	60.13	53.32	52.19	59.19	52.76	53.10
Hindi	56.32	59.76	57.17	55.39	58.44	55.65
Japanese	41.13	44.95	40.13	42.43	43.76	41.18
Korean	45.78	42.99	43.04	44.78	40.22	41.14

Table 37. Cross-lingual performance (F1 scores) of **FourIE** when it is trained on English training data and evaluated on test data of other languages in MEE.

hyper-parameters for the pipeline model include: 2 hidden layers with 250 hidden units in each layer for the feed-forward networks, 8 for mini-batch size, and $1e$-2 for learning rate with the Adam optimizer. For the joint IE models, we utilize the same hyper-parameters suggested in the original papers, i.e., OneIE Y. Lin et al. (2020) and FourIE M. V. Nguyen, Lai, and Nguyen (2021b).

Results: The results for monolingual experiments over different languages in MEE are presented in Tables 35 and 36 (i.e., with mBERT and XLM-RoBERTa encoders respectively). There are several observations from the tables. First, the models' performance on individual languages and on average for all three tasks EMD, ED, and EAE is still far from being perfect (i.e., all average performance is less than 69%), thus indicating considerable challenges in our multilingual EE dataset for future research. In addition, comparing the current state-of-the-art joint IE model (i.e., FourIE) with the pipeline method, we find that FourIE is better than the pipeline model on average, especially for the EAE task with significant performance gap. As such, we attribute this to the ability of joint models to mitigate error propagation to EAE from EMD and ED to boost the performance. In addition to the joint modeling of tasks, the superiority of FourIE can be attributed to

Language	Entity	Event	Argument
English Devlin et al. (2019b)	70.21	73.18	66.19
Spanish Cañete et al. (2020)	67.29	65.14	60.13
Portuguese Souza, Nogueira, and Lotufo (2020)	70.21	68.88	67.13
Polish Kłeczek (2021)	65.78	61.23	59.14
Turkish MDZ (2021)	67.34	64.19	58.72

Table 38. Test data performance (F1) of **FourIE** in monolingual learning using available language-specific BERT models on MEE. The citations indicate the sources of the language-specific models.

its use of structureal information. Due to its best average performance, FourIE will be leveraged in our next experiments. Finally, we find that XLM-RoBERTa generally has better performance than mBERT (i.e., on average) for EE models. Future research can thus focus on XLM-RoBERTa to develop better EE models for multilingual settings.

Cross-lingual Evaluation: To further understand the cross-lingual generalization challenges in MEE, Table 37 reports the performance of FourIE in the cross-lingual transfer learning settings where the model is trained on English training data (source language) and tested on test data of the other languages in MEE. As can be seen, compared to performance on English test set, FourIE suffers from significant performance drops over different tasks and multilingual encoders when it is evaluated on other languages. It thus demonstrates inherent challenges of cross-lingual generalization for complete EE models that can be further studied with MEE. In addition, the performance loss due to cross-lingual testing varies across different target languages (e.g., 10.88% loss for Spanish vs. 33.42% loss for Japanese in EAE task). These variations can be attributed to different levels of divergence between languages (e.g., sentence structures and morphology) that hinder cross-lingual knowledge transfer for EE.

Language	Entity	Event	Argument
English Y. Liu et al. (2019)	70.32	72.28	69.19
Spanish MMG (2021)	70.23	61.34	60.28
Polish CLARIN-PL (2021)	68.12	60.89	60.34
Hindi Parmar (2021)	64.91	59.09	60.38
Japanese Wongso (2021)	69.72	60.45	71.45

Table 39. Test data performance (F1) of **FourIE** in monolingual learning using available language-specific RoBERTa models on MEE. The citations indicate the sources of the language-specific models.

Language-Specific Encoders: To study the effectiveness of pre-trained language models as text encoders for EE models, we compare the performance of FourIE when the multilingual encoders mBERT or XLM-RoBERTa are replaced with comparable language-specific encoders (i.e., BERT-based models for mBERT and RoBERTa-based models for XLM-RoBERTa). Using publicly available pre-trained language models for our languages in MEE, Tables 38 and 39 show the monolingual performance over test data of the languages for BERT-based and RoBERTa-based models (respectively). Comparing corresponding performance in Tables 35, 36, 38 and 39, it is clear that language-specific language models all under-perform their multilingual counterparts over different EE tasks and languages, thus suggesting the benefits of multilingual data to train language model encoders to boost EE performance over different languages.

Source Language Impact: Finally, to study the impact of the source language for cross-lingual transfer learning for EE, we compare the performance of FourIE when either English or another comparable language is used as the source language to provide training data to train the model. In particular, we choose Polish as a comparable language for English as it has the same sentence structure (i.e., both languages have Subject-Verb-Object order) and entails similar density and type distributions for entity/event mentions as English. Table 40 shows the performance

Language	Trained on English			Trained on Polish		
	Entity	Event	Argument	Entity	Event	Argument
Portuguese	53.22	50.79	40.21	54.17	51.70	42.33
Spanish	50.76	43.72	41.16	51.72	45.22	45.81
Turkish	54.44	50.12	55.71	53.99	50.78	56.15
Hindi	51.44	52.78	45.27	52.21	53.00	47.24
Japanese	36.16	41.23	38.13	37.17	40.13	39.28
Korean	43.72	40.08	37.29	42.08	39.78	37.10

Table 40. Cross-lingual performance (F1) of **FourIE** with XLM-RoBERTa encoder when it is trained on English or Polish training data, and tested on test data of the other languages in MEE. We use 3,500 random annotated segments from the training sets of English and Polish to train the model.

of the models when they are tested over test data of the other 6 languages in MEE. Here, to make it comparable, we use the same number of annotated segments (i.e., 3,500) sampled from training data of English and Polish to train the FourIE model. Interestingly, we find that Polish can lead to better performance for FourIE than English over a majority of task and target language pairs (i.e., over 4 languages for EMD and ED, and 5 languages for EAE). A possible explanation for this issue comes from richer event patterns that Polish might introduce to produce allow better cross-lingual generalization for EE than those for English. As such, this superior performance of Polish challenges the common practice of using English as the source language in cross-lingual transfer learning studies for EE and NLP. Future research can explore this direction to better understand the differences between languages to best select a source language to optimize performance over a target language for EE.

5.3 Related Works

Due to its importance, various datasets have been recently developed for EE in different domains, including CySecED Man, Trong Le, Pouran Ben Veyseh, Nguyen, and Nguyen (2020) (for cybersecurity domain), LitBank (for literacy) Sims,

Park, and Bamman (2019b), MAVEN X. Wang et al. (2020), RAMS Ebner et al. (2020), and WikiEvents (for Wikipedia texts). However, these datasets are only annotated for English texts. There exist several multilingual datasets for EE, ACE Walker et al. (2006a), TAC KBP Mitamura et al. (2016, 2017), and TempEval-2 Verhagen et al. (2010); however, such datasets only provide annotation for a handful of popular languages with limited number of event mentions and might not fully support all EE tasks (e.g., missing EAE in TAC KBP and TempEval-2).

Regarding model development, existing EE methods can be categorized into feature-based Ahn (2006); Hong et al. (2011b); Ji and Grishman (2008); Q. Li et al. (2013); Liao and Grishman (2010a); B. Yang and Mitchell (2016b) or deep learning Y. Chen et al. (2015); Y. Lin et al. (2020); T. H. Nguyen, Cho, and Grishman (2016); Pouran Ben Veyseh, Lai, Dernoncourt, and Nguyen (2021); Pouran Ben Veyseh, Nguyen, Ngo Trung, Min, and Nguyen (2021); Sha et al. (2018); X. Wang, Han, et al. (2019b) methods. While most prior EE methods have been designed for one popular language, there have been growing interests in multilingual and cross-lingual learning for EE in recent work, featuring multilingual PLMs (i.e., mBERT and XLMR) as the key component for representation learning Ahmad et al. (2020); Z. Chen and Ji (2009a); M'hamdi, Freedman, and May (2019); M. V. Nguyen, Nguyen, Min, and Nguyen (2021). However, as such works only rely on existing multilingual EE datasets, their evaluation is limited to a few popular languages and fails to evaluate the generalization over many other languages.

5.4 Conclusion

We present a novel multilingual EE dataset, i.e., MEE, that covers 8 typologically different languages with more than 50K event mentions to support training of large deep learning models. MEE provides complete annotation for

three EE sub-tasks, i.e., entity mention detection, event detection, and event argument extraction. To study the challenges in MEE, we conduct extensive analysis and experiments with different EE methods in the monolingual and cross-lingual learning settings. Our results demonstrate various challenges for EE in the multilingual settings that can be further pursued with MEE. Moreover, we demonstrate that structure-based models, e.g., FourIE, outperform their sequential counterparts.

CHAPTER VI

CONCLUSION

Information Extraction is one of the essential tasks in Natural Language Processing. The objective of this field is to automate the extraction of meaningful information from raw text and provide them in a more structured format (e.g., a knowledge base). To this end, several sub-tasks should be solved to complete a pipeline of IE. These tasks range from entity recognition, entity linking, relation extraction, event detection, and event argument extraction. Moreover, each of these tasks may have several settings and sub-tasks (e.g., event co-reference detection to identify the event mentions that refer to the same event in the text). In the literature, each of these tasks and sub-tasks has a rich body of research. One of the directions that have shown promising results in various IE tasks is structure-based models. Structural information for information extraction models refers to any interactions between different components of the input text (i.e., words, phrases, or sentences). For instance, a syntactic tree which is commonly modeled by dependency trees could be used to encode the syntactical interactions between the words, so that those parts of the input text that are syntactically related to each other would have an influence on their representations obtained by the model. The structural information could be used to overcome the long distances between related words/sentences that are difficult to encode with sequential models.

Due to the importance of structural information for IE, in this dissertation, we study the effective directions for incorporating various types of structures into IE models. In particular, this dissertation category the methods into three general approaches: (1) Models that use the existing structures, i.e. structures obtained from pre-trained parsers; (2) Models that infer a task-specific structure for the

input text; (3) Models that in infer structure between samples in a batch of data. For each of these categories, we present novel methods that improve the state-of-the-art performance on selected IE tasks. Finally, in order to show that the structure-based models could be helpful for less-explored IE settings, in the final Chapter, we present a novel multilingual event argument extraction dataset and we show that a structure-based model outperforms other sequential models.

For the models that employ the existing structures, we study the task of document-level event argument extraction. For this task, the objective is to identify the role of entities in a document in a specific event mentioned in the text. For this task, we introduced a novel method to incorporate the structural information of different types, e.g., syntactic, semantic, and external knowledge. The proposed method demonstrates an effective application of graph transformer to this end. The presented model achieves state-of-the-art performance for this setting of EAE.

Next, to show how the structural information could be inferred, for the task of relation extraction, we show that a special architecture, i.e., ordered neuron LSTM (ON-LSTM), can be utilized to induce semantic structure for the input text. Moreover, we present a novel mechanism based on mutual information that can be used to ensure the consistency of the inferred semantic structure with syntactic information.

For the final category of structure-based models, we study how the interactions between samples in a batch of data can be utilized for IE models. In particular, we first discuss the necessity of generating synthetic data to address data scarcity for the settings that suffer from the small size of training datasets. Next, as a mechanism to ensure that the generated synthetic data have high quality, we propose a novel solution based on optimal transport to obtain a bipartite

graph between original and synthetic data. This graph is employed to compute the consistency between the groups of samples.

Finally, to showcase that structural information is helpful for less-explored settings of IE, in the final Chapter, we study how structure-based models can be used for cross-lingual event extraction. In particular, our experiments show that joint models that employ structural information, i.e., instance interactions in FourIE, outperform other sequential models such as BERT. This finding shows that graph-based models could be a remedy for the challenges of training a deep model in low-resource settings.

While this dissertation provides a broad overview of different methods to employ structures for IE, there are directions that are left for future research. One potential direction is to study how the structures can be used to integrate IE models with other downstream applications such as Question Answering and Summarization. As discussed in this dissertation, task-specific structures are a powerful tool to improve the performance of the task at hand. However, for multitasking setting the interactions between structures used in each task are not studied here. Another direction to explore in future works is to evaluate the effectiveness of the proposed solutions for informal settings. For instance, structures in conversations or transcripts could be drastically different from the structures of the formal text, and more research on this is needed. Finally, exploring the application of the proposed technique in other tasks, e.g., document classification, natural language inference, etc, is another potential direction to consider for future work.

REFERENCES CITED

Aguilar, G., & Solorio, T. (2019). Dependency-aware named entity recognition with relative and global attentions. *arXiv preprint arXiv:1909.05166*.

Ahmad, W. U., Peng, N., & Chang, K.-W. (2020). Gate: Graph attention transformer encoder for cross-lingual relation and event extraction. *arXiv preprint arXiv:2010.03009*.

Ahn, D. (2006). The stages of event extraction. In *Proceedings of the workshop on annotating and reasoning about time and events* (pp. 1–8).

Amir Pouran Ben Veyseh, T. H. N., Tuan Ngo Nguyen. (2020). *Graph transformer networks with syntactic and semantic structuresfor event argument extraction.*

Anaby-Tavor, A., Carmeli, B., Goldbraich, E., Kantor, A., Kour, G., Shlomov, S., ... Zwerdling, N. (2020). Do not have enough data? deep learning to the rescue! In *Proceedings of the association for the advancement of artificial intelligence (aaai).*

Andreas, J., Rohrbach, M., Darrell, T., & Klein, D. (2016). Neural module networks. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 39–48).

Araki, J., & Mitamura, T. (2018). Open-domain event detection using distant supervision. In *Proceedings of the international conference on computational linguistics (coling).*

Attardi, G. (2015). *Wikiextractor.* `https://github.com/attardi/wikiextractor`. GitHub.

Belghazi, M. I., Baratin, A., Rajeswar, S., Ozair, S., Bengio, Y., Courville, A., & Hjelm, R. D. (2018). Mine: mutual information neural estimation. In *Icml.*

Björkelund, A., & Kuhn, J. (2014). Learning structured perceptrons for coreference resolution with latent antecedents and non-local features. In *Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 47–57).

Björne, J., & Salakoski, T. (2018, July). Biomedical event extraction using convolutional neural networks and dependency parsing. In *Proceedings of the BioNLP 2018 workshop* (pp. 98–108). Melbourne, Australia: Association for Computational Linguistics. Retrieved from `https://www.aclweb.org/anthology/W18-2311` doi: 10.18653/v1/W18-2311

Blevins, T., & Zettlemoyer, L. (2020). Moving down the long tail of word sense disambiguation with gloss informed bi-encoders. In *Acl.*

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., & Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems* (pp. 2787–2795).

Bosselut, A., Rashkin, H., Sap, M., Malaviya, C., Celikyilmaz, A., & Choi, Y. (2019). Comet: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the annual meeting of the association for computational linguistics (acl).*

Bunescu, R., & Mooney, R. (2005). A shortest path dependency kernel for relation extraction. In *Proceedings of human language technology conference and conference on empirical methods in natural language processing* (pp. 724–731).

Cao, Y., Hou, L., Li, J., & Liu, Z. (2018). Neural collective entity linking. *arXiv preprint arXiv:1811.08603*.

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 international conference on learning representations (iclr) 2020.*

Chan, Y. S., & Roth, D. (2010). Exploiting background knowledge for relation extraction. In *Coling.*

Chen, Y., Xu, L., Liu, K., Zeng, D., & Zhao, J. (2015). Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the annual meeting of the association for computational linguistics (acl).*

Chen, Z., & Ji, H. (2009a). 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.*

Chen, Z., & Ji, H. (2009b). Language specific issue and feature exploration in chinese event extraction. In *Proceedings of human language technologies: The 2009 annual conference of the north american chapter of the association for computational linguistics, companion volume: Short papers* (pp. 209–212).

Christopoulou, F., Miwa, M., & Ananiadou, S. (2019). Connecting the dots: Document-level neural relation extraction with edge-oriented graphs. In *Emnlp.*

CLARIN-PL, C.-P. (2021). Polish roberta.. Retrieved from `https://huggingface.co/clarin-pl/roberta-polish-kgr10`

Clark, K., & Manning, C. D. (2016). Improving coreference resolution by learning entity-level distributed representations. *arXiv preprint arXiv:1606.01323*.

Collins, M., & Singer, Y. (1999). Unsupervised models for named entity classification. In *1999 joint sigdat conference on empirical methods in natural language processing and very large corpora.*

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., ... Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th annual meeting of the 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). Event detection with relation-aware graph convolutional neural networks.

Dernoncourt, F., Lee, J. Y., & Szolovits, P. (2017). Neuroner: an easy-to-use program for named-entity recognition based on neural networks. *arXiv preprint arXiv:1705.05487*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019a). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Naacl-hlt.*

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019b). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Emnlp.*

Doddington, G. R., Mitchell, A., Przybocki, M. A., Ramshaw, L. A., Strassel, S. M., & Weischedel, R. M. (2004). The automatic content extraction (ace) program-tasks, data, and evaluation. In *Lrec* (Vol. 2, pp. 837–840).

dos Santos, C., Xiang, B., & Zhou, B. (2015). Classifying relations by ranking with convolutional neural networks. In *Acl.*

Duan, S., He, R., & Zhao, W. (2017). Exploiting document level information to improve event detection via recurrent neural networks. In *Proceedings of the international joint conference on natural language processing (ijcnlp).*

Ebner, S., Xia, P., Culkin, R., Rawlins, K., & Van Durme, B. (2019). Multi-sentence argument linking. *arXiv preprint arXiv:1911.03766*.

Ebner, S., Xia, P., Culkin, R., Rawlins, K., & Van Durme, B. (2020). Multi-sentence argument linking. In *Proceedings of the annual meeting of the association for computational linguistics (acl).*

Eftimov, T., Koroušić Seljak, B., & Korošec, P. (2017). A rule-based named-entity recognition method for knowledge extraction of evidence-based dietary recommendations. *PloS one*, *12*(6), e0179488.

Fang, K., & Jian, F. (2019). Incorporating structural information for better coreference resolution. In *Proceedings of the 28th international joint conference on artificial intelligence* (pp. 5039–5045).

Fang, Z., Cao, Y., Li, Q., Zhang, D., Zhang, Z., & Liu, Y. (2019). Joint entity linking with deep reinforcement learning. In *The world wide web conference* (pp. 438–447).

Fang, Z., Cao, Y., Li, R., Zhang, Z., Liu, Y., & Wang, S. (2020). High quality candidate generation and sequential graph attention network for entity linking. In *Proceedings of the web conference 2020* (pp. 640–650).

Fei, H., Li, X., Li, D., & Li, P. (2019). End-to-end deep reinforcement learning based coreference resolution. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 660–665).

Feizabadi, P. S., & Padó, S. (2015). Combining seemingly incompatible corpora for implicit semantic role labeling. In *The fourth joint conference on lexical and computational semantics.*

Feng, J., Huang, M., Zhao, L., Yang, Y., & Zhu, X. (2018). Reinforcement learning for relation classification from noisy data. *Proceedings of the Thirty-Second AAAI Con- ference on Artificial Intelligence*.

Ferguson, J., Lockard, C., Weld, D. S., & Hajishirzi, H. (2018). Semi-supervised event extraction with paraphrase clusters. In *Proceedings of the conference of the north american chapter of the association for computational linguistics: Human language technologies (naacl-hlt).*

Fu, L., Nguyen, T. H., Min, B., & Grishman, R. (2017). Domain adaptation for relation extraction with domain adversarial neural network. In *Ijcnlp.*

Fu, T.-J., Li, P.-H., & Ma, W.-Y. (2019). Graphrel: Modeling text as relational graphs for joint entity and relation extraction. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 1409–1418).

Ganea, O.-E., & Hofmann, T. (2017). Deep joint entity disambiguation with local neural attention. *arXiv preprint arXiv:1704.04920*.

Gardent, C., Shimorina, A., Narayan, S., & Perez-Beltrachini, L. (2017). Creating training corpora for nlg micro-planning. In *55th annual meeting of the association for computational linguistics (acl).*

Gerber, M., & Chai, J. Y. (2012). Semantic role labeling of implicit arguments for nominal predicates. In *Computational linguistics.*

Gorinski, P., Ruppenhofer, J., & Sporleder, C. (2013). Towards weakly supervised resolution of null instantiations. In *Iwcs.*

Guo, Z., Zhang, Y., & Lu, W. (2019). Attention guided graph convolutional networks for relation extraction. In *Acl.*

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).

Hachey, B., Radford, W., Nothman, J., Honnibal, M., & Curran, J. R. (2013). Evaluating entity linking with wikipedia. *Artificial intelligence*, *194*, 130–150.

Hancock, B., Bringmann, M., Varma, P., Liang, P., Wang, S., & Ré, C. (2018). Training classifiers with natural language explanations. In *Acl.*

Hendrickx, I., Kim, S. N., Kozareva, Z., Nakov, P., Ó Séaghdha, D., Padó, S., ... Szpakowicz, S. (2010). Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *the 5th international workshop on semantic evaluation.*

Hendrickx, I., Kim, S. N., Kozareva, Z., Nakov, P., Séaghdha, D. O., Padó, S., ... Szpakowicz, S. (2009). Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals..

Hieu Man Duc Trong, A. P. B. V. T. H. N., Duc Trong Le. (2020). Introducing a new dataset for event detection in cybersecurity texts. In *Emnlp.*

Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. In *Proceedings of the deep learning workshop at neurips.*

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Hong, Y., Zhang, J., Ma, B., Yao, J., Zhou, G., & Zhu, Q. (2011a). Using cross-entity inference to improve event extraction. In *Proceedings of the annual meeting of the association for computational linguistics (acl).*

Hong, Y., Zhang, J., Ma, B., Yao, J., Zhou, G., & Zhu, Q. (2011b). Using cross-entity inference to improve event extraction. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1* (pp. 1127–1136).

Hu, L., Zhang, L., Shi, C., Nie, L., Guan, W., & Yang, C. (2019). Improving distantly-supervised relation extraction with joint label embedding. 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. 3812–3820).

Huang, B., Ou, Y., & Carley, K. M. (2018). Aspect level sentiment classification with attention-over-attention neural networks. In *Sbp-brims.*

Huang, L., Cassidy, T., Feng, X., Ji, H., Voss, C., Han, J., & Sil, A. (2016). Liberal event extraction and event schema induction. In *Proceedings of the annual meeting of the association for computational linguistics (acl).*

Huang, R., & Riloff, E. (2012). Bootstrapped training of event extraction classifiers. In *Proceedings of the conference of the european chapter of the association for computational linguistics (eacl).*

Ji, H., & Grishman, R. (2008). Refining event extraction through cross-document inference. In *Acl.*

Ji, H., & Grishman, R. (2011). Knowledge base population: Successful approaches and challenges. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies* (pp. 1148–1158).

Jie, Z., & Lu, W. (2019). Dependency-guided lstm-crf for named entity recognition. *arXiv preprint arXiv:1909.10148*.

Keith, K., Handler, A., Pinkham, M., Magliozzi, C., McDuffie, J., & O'Connor, B. (2017). Identifying civilians killed by police with distantly supervised entity-event extraction. In *Proceedings of the conference on empirical methods in natural language processing (emnlp).*

Khalife, S., & Vazirgiannis, M. (2018). Scalable graph-based individual named entity identification. *arXiv preprint arXiv:1811.10547*.

Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Krippendorff, K. (2011). Computing krippendorff's alpha-reliability..

Kumar, V., Choudhary, A., & Cho, E. (2020). Data augmentation using pre-trained transformer models. *arXiv preprint arXiv:2003.02245*.

Kłeczek, D. (2021). Polish bert.. Retrieved from https://huggingface.co/dkleczek/bert-base-polish-uncased-v1

Lai, V. D., Dernoncourt, F., & Nguyen, T. H. (2020a). Exploiting the matching information in the support set for few shot event classification. In *Proceedings of the 24th pacific-asia conference on knowledge discovery and data mining (pakdd).*

Lai, V. D., Dernoncourt, F., & Nguyen, T. H. (2020b). Extensively matching for few-shot learning event detection. In *Proceedings of the 1st joint workshop on narrative understanding, storylines, and events (nuse) at acl 2020.*

Lai, V. D., Nguyen, T. N., & Nguyen, T. H. (2020a). Event detection: Gate diversity and syntactic importance scores for graph convolution neural networks. In *Emnlp.*

Lai, V. D., Nguyen, T. N., & Nguyen, T. H. (2020b). Event detection: Gate diversity and syntactic importance scores for graph convolution neural networks. In *Proceedings of the conference on empirical methods in natural language processing (emnlp).*

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Laparra, E., & Rigau, G. (2013). Sources of evidence for implicit argument resolution. In *Iwcs.*

Lappin, S., & Leass, H. J. (1994). An algorithm for pronominal anaphora resolution. *Computational linguistics*, *20*(4), 535–561.

Le, D. M., & Nguyen, T. H. (2021). Fine-grained event trigger detection. In *Eacl.*

Le, M., & Fokkens, A. (2018). Neural models of selectional preferences for implicit semantic role labeling. In *The eleventh international conference on language resources and evaluation.*

Le, P., & Titov, I. (2019). Distant learning for entity linking with automatic noise detection. *arXiv preprint arXiv:1905.07189*.

Lee, K., He, L., Lewis, M., & Zettlemoyer, L. (2017). End-to-end neural coreference resolution. *arXiv preprint arXiv:1707.07045*.

Lee, K., He, L., & Zettlemoyer, L. (2018). Higher-order coreference resolution with coarse-to-fine inference. In *Naacl-hlt.*

Li, D., Huang, L., Ji, H., & Han, J. (2019). 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).

Li, J., Sun, A., Han, J., & Li, C. (2020). A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*.

Li, J., Sun, Y., Johnson, R. J., Sciaky, D., Wei, C.-H., Leaman, R., ... Lu, Z. (2016). Biocreative v cdr task corpus: a resource for chemical disease relation extraction. *Database*, *2016*.

Li, M., Zareian, A., Zeng, Q., Whitehead, S., Lu, D., Ji, H., & Chang, S.-F. (2020). Cross-media structured common space for multimedia event extraction. *arXiv preprint arXiv:2005.02472*.

Li, M., Zeng, Q., Lin, Y., Cho, K., Ji, H., May, J., ... Voss, C. (2020). Connecting the dots: Event graph schema induction with path language modeling. In *Proc. the 2020 conference on empirical methods in natural language processing (emnlp2020).*

Li, Q., Ji, H., & Huang, L. (2013). Joint event extraction via structured prediction with global features. In *Proceedings of the annual meeting of the association for computational linguistics (acl).*

Li, W., Cheng, D., He, L., Wang, Y., & Jin, X. (2019). Joint event extraction based on hierarchical event schemas from framenet. *IEEE Access*, *7*, 25001–25015.

Liao, S., & Grishman, R. (2010a). Filtered ranking for bootstrapping in event extraction. In *Coling.*

Liao, S., & Grishman, R. (2010b). Using document level cross-event inference to improve event extraction. In *Acl.*

Lin, T., Etzioni, O., et al. (2012). Entity linking at web scale. In *Proceedings of the joint workshop on automatic knowledge base construction and web-scale knowledge extraction (akbc-wekex)* (pp. 84–88).

Lin, Y., Ji, H., Huang, F., & Wu, L. (2020). A joint neural model for information extraction with global features. In *Proceedings of the 58th annual meeting of the association for computational linguistics.*

Lin, Y., Shen, S., Liu, Z., Luan, H., & Sun, M. (2016). Neural relation extraction with selective attention over instances. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 2124–2133).

Liu, J., Chen, Y., & Liu, K. (2019). Exploiting the ground-truth: An adversarial imitation based knowledge distillation approach for event detection. In *Proceedings of the association for the advancement of artificial intelligence (aaai).*

Liu, J., Chen, Y., Liu, K., & Zhao, J. (2018). Event detection via gated multilingual attention mechanism. In *Aaai.*

Liu, S., Chen, Y., Liu, K., & Zhao, J. (2017). Exploiting argument information to improve event detection via supervised attention mechanisms. In *Acl.*

Liu, T., Wang, K., Chang, B., & Sui, Z. (2017). A soft-label method for noise-tolerant distantly supervised relation extraction. In *Proceedings of the 2017 conference on empirical methods in natural language processing* (pp. 1790–1795).

Liu, X., Li, Y., Wu, H., Zhou, M., Wei, F., & Lu, Y. (2013). Entity linking for tweets. In *Proceedings of the 51st annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 1304–1311).

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

Liu, Y., Wei, F., Li, S., Ji, H., Zhou, M., & Wang, H. (2015). A dependency-based neural network for relation classification. In *Acl.*

Lu, Y., Lin, H., Han, X., & Sun, L. (2019). Distilling discrimination and generalization knowledge for event detection via delta-representation learning. In *Proceedings of the annual meeting of the association for computational linguistics (acl).*

Luan, Y., He, L., Ostendorf, M., & Hajishirzi, H. (2018). Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Emnlp.*

Ma, C., Tamura, A., Utiyama, M., Sumita, E., & Zhao, T. (2019). Improving neural machine translation with neural syntactic distance. 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. 2032–2037).

Madaan, A., Rajagopal, D., Yang, Y., Ravichander, A., Hovy, E., & Prabhumoye, S. (2020). Eigen: Event influence generation using pre-trained language models. *arXiv preprint arXiv:2010.11764*.

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

Man Duc Trong, H., Trong Le, D., Pouran Ben Veyseh, A., Nguyen, T., & Nguyen, T. H. (2020). Introducing a new dataset for event detection in cybersecurity texts. In *Proceedings of the conference on empirical methods in natural language processing (emnlp).*

Mani, I., Bloedorn, E., & Gates, B. (1998). Using cohesion and coherence models for text summarization. In *Intelligent text summarization symposium* (pp. 69–76).

McClosky, D., Surdeanu, M., & Manning, C. (2011). Event extraction as dependency parsing. In *Bionlp shared task workshop.*

MDZ, D. L. t. (2021). Turkish bert.. Retrieved from `https://huggingface.co/dbmdz/bert-base-turkish-uncased`

M'hamdi, M., Freedman, M., & May, J. (2019). Contextualized cross-lingual event trigger extraction with minimal resources. In *Proceedings of the 23rd conference on computational natural language learning (conll).*

Mikheev, A., Moens, M., & Grover, C. (1999). Named entity recognition without gazetteers. In *Ninth conference of the european chapter of the association for computational linguistics.*

Min, B., Shi, S., Grishman, R., & Lin, C.-Y. (2012). Ensemble semantics for large-scale unsupervised relation extraction. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning* (pp. 1027–1037).

Mintz, M., Bills, S., Snow, R., & Jurafsky, D. (2009). 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).

Mitamura, T., Liu, Z., & Hovy, E. H. (2016). Overview of TAC-KBP 2016 event nugget track. In *Proceedings of the text analysis conference (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 *Proceedings of the text analysis conference (tac).*

Miwa, M., & Bansal, M. (2016). End-to-end relation extraction using lstms on sequences and tree structures. In *Acl.*

Miwa, M., Thompson, P., Korkontzelos, I., & Ananiadou, S. (2014). Comparable study of event extraction in newswire and biomedical domains. In *Coling.*

MMG, M. (2021). Spanish roberta.. Retrieved from `https://huggingface.co/MMG/mlm-spanish-roberta-base`

Nadeau, D., & Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, *30*(1), 3–26.

Naik, A., & Rosé, C. (2020). Towards open domain event trigger identification using adversarial domain adaptation. *arXiv preprint arXiv:2005.11355*.

Nan, G., Guo, Z., Sekulic, I., & Lu, W. (2020). Reasoning with latent structure refinement for document-level relation extraction. In *Acl.*

Nédellec, C., Bossy, R., Kim, J.-D., Kim, J.-J., Ohta, T., Pyysalo, S., & Zweigenbaum, P. (2013). Overview of bionlp shared task 2013. In *Proceedings of the bionlp shared task 2013 workshop* (pp. 1–7).

Ng, V., & Cardie, C. (2002). Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th annual meeting of the association for computational linguistics* (pp. 104–111).

Ngo, N., Nguyen, T. N., & Nguyen, T. H. (2020). Learning to select important context words for event detection. In *Proceedings of the 24th pacific-asia conference on knowledge discovery and data mining (pakdd).*

Nguyen, L. T., Van Ngo, L., Than, K., & Nguyen, T. H. (2019). Employing the correspondence of relations and connectives to identify implicit discourse relations via label embeddings. In *Proceedings of the annual meeting of the association for computational linguistics (acl).*

Nguyen, M., & Nguyen, T. H. (2018b). Who is killed by police: Introducing supervised attention for hierarchical lstms. In *Coling.*

Nguyen, M. V., Lai, V. D., & Nguyen, T. H. (2021a). Cross-task instance representation interactions and label dependencies for joint information extraction with graph convolutional networks. In *Proceedings of the conference of the north american chapter of the association for computational linguistics: Human language technologies (naacl-hlt).*

Nguyen, M. V., Lai, V. D., & Nguyen, T. H. (2021b). Cross-task instance representation interactions and label dependencies for joint information extraction with graph convolutional networks. In *Proceedings of the conference of the north american chapter of the association for computational linguistics: Human language technologies (naacl-hlt).*

Nguyen, M. V., Lai, V. D., Veyseh, A. P. B., & Nguyen, T. H. (2021). 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.*

Nguyen, M. V., Nguyen, T. N., Min, B., & Nguyen, T. H. (2021, November). Crosslingual transfer learning for relation and event extraction via word category and class alignments. In *Proceedings of the 2021 conference on empirical methods in natural language processing* (pp. 5414–5426). Online and Punta Cana, Dominican Republic: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2021.emnlp-main.440`

Nguyen, T. H., Cho, K., & Grishman, R. (2016). Joint event extraction via recurrent neural networks. In *Naacl-hlt.*

Nguyen, T. H., Cho, K., & Grishman, R. (2016a). Joint event extraction via recurrent neural networks. In *Naacl.*

Nguyen, T. H., & Grishman, R. (2014). Employing word representations and regularization for domain adaptation of relation extraction. In *Acl.*

Nguyen, T. H., & Grishman, R. (2015a). Event detection and domain adaptation with convolutional neural networks. In *Acl-ijcnlp.*

Nguyen, T. H., & Grishman, R. (2015b). 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).* Retrieved from `https://aclanthology.org/P15-2060` doi: 10.3115/v1/P15-2060

Nguyen, T. H., & Grishman, R. (2015c). Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 1st workshop on vector space modeling for natural language processing* (pp. 39–48).

Nguyen, T. H., & Grishman, R. (2015a). Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 1st naacl workshop on vector space modeling for nlp (vsm).*

Nguyen, T. H., & Grishman, R. (2016). Combining neural networks and log-linear models to improve relation extraction. In *Proceedings of ijcai workshop on deep learning for artificial intelligence.*

Nguyen, T. H., & Grishman, R. (2018). Graph convolutional networks with argument-aware pooling for event detection. In *Aaai.*

Nguyen, T. H., & Grishman, R. (2018a). Graph convolutional networks with argument-aware pooling for event detection. In *Aaai.*

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. H., Plank, B., & Grishman, R. (2015c). Semantic representations for domain adaptation: A case study on the tree kernel-based method for relation extraction. In *Acl-ijcnlp.*

Nguyen, T. H., Sil, A., Dinu, G., & Florian, R. (2016). Toward mention detection robustness with recurrent neural networks. *arXiv preprint arXiv:1602.07749*.

Nguyen, T. H., Sil, A., Dinu, G., & Florian, R. (2016b). Toward mention detection robustness with recurrent neural networks. In *Proceedings of ijcai workshop on deep learning for artificial intelligence (dlai).*

Nguyen, T. M., & Nguyen, T. H. (2019a). One for all: Neural joint modeling of entities and events. In *Proceedings of the association for the advancement of artificial intelligence (aaai).*

Nguyen, T. M., & Nguyen, T. H. (2019b). One for all: Neural joint modeling of entities and events. In *Aaai.*

Nguyen, T.-V. T., & Moschitti, A. (2011). End-to-end relation extraction using distant supervision from external semantic repositories. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies* (pp. 277–282).

Papanikolaou, Y., & Pierleoni, A. (2020). Dare: Data augmented relation extraction with gpt-2. In *Scinlp workshop at the conference on automated knowledge base construction (akbc).*

Parmar, S. (2021). Hindi roberta.. Retrieved from `https://huggingface.co/surajp/RoBERTa-hindi-guj-san`

Passonneau, R. (2006). Measuring agreement on set-valued items (masi) for semantic and pragmatic annotation. In *Lrec.*

Patwardhan, S., & Riloff, E. (2009). A unified model of phrasal and sentential evidence for information extraction. In *Emnlp.*

Peng, B., Zhu, C., Zeng, M., & Gao, J. (2020). Data augmentation for spoken language understanding via pretrained models. *arXiv preprint arXiv:2004.13952*.

Peng, N., Poon, H., Quirk, C., Toutanova, K., & Yih, W.-t. (2017). Cross-sentence n-ary relation extraction with graph lstms. In *Tacl.*

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

Plank, B. (2011). Domain adaptation for parsing. In *Ph.d. thesis. university of groningen.*

Pouran Ben Veyseh, A., 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

Pouran Ben Veyseh, A., Nguyen, M. V., Ngo Trung, N., Min, B., & Nguyen, T. H. (2021). Modeling document-level context for event detection via important context selection. In *Proceedings of the 2021 conference on empirical methods in natural language processing.*

Pouran Ben Veyseh, A., Nguyen, T. N., & Nguyen, T. H. (2020). Graph transformer networks with syntactic and semantic structures for event argument extraction. In *Emnlp findings.*

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

Qin, P., Xu, W., & Wang, W. Y. (2018). Robust distant supervision relation extraction via deep reinforcement learning. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics.*.

Qiu, D., Zhang, Y., Feng, X., Liao, X., Jiang, W., Lyu, Y., ... Zhao, J. (2019). Machine reading comprehension using structural knowledge graph-aware network. 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. 5898–5903).

Quirk, C., & Poon, H. (2016). Distant supervision for relation extraction beyond the sentence boundary. *arXiv preprint arXiv:1609.04873*.

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

Raghunathan, K., Lee, H., Rangarajan, S., Chambers, N., Surdeanu, M., Jurafsky, D., & Manning, C. D. (2010). A multi-pass sieve for coreference resolution. In *Proceedings of the 2010 conference on empirical methods in natural language processing* (pp. 492–501).

Rahman, A., & Ng, V. (2009). Supervised models for coreference resolution. In *Proceedings of the 2009 conference on empirical methods in natural language processing* (pp. 968–977).

Rao, S., Marcu, D., Knight, K., & Daumé III, H. (2017). Biomedical event extraction using abstract meaning representation. In *Bionlp 2017* (pp. 126–135).

Ratinov, L., & Roth, D. (2012). Learning-based multi-sieve co-reference resolution with knowledge. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning* (pp. 1234–1244).

Rich ere annotation guidelines overview. (2016). In *Linguistic data consortium,philadelphia.*

Riedel, S., & McCallum, A. (2011). Robust biomedical event extraction with dual decomposition and minimal domain adaptation. In *Bionlp shared task 2011 workshop.*

Riloff, E., & Shoen, J. (1995). Automatically acquiring conceptual patterns without an annotated corpus. In *Third workshop on very large corpora.*

Ritter, A., Etzioni, O., & Clark, S. (2012). Open domain event extraction from twitter. In *Proceedings of the 18th acm sigkdd international conference on knowledge discovery and data mining* (pp. 1104–1112).

Ruppenhofer, J., Sporleder, C., Morante, R., Baker, C. F., & Palmer, M. (2010). Semeval-2010 task 10: Linking events and their participants in discourse. In *Proceedings of the 5th international workshop on semantic evaluation, acl.*

Saha, S., Majumder, A., Hasanuzzaman, M., & Ekbal, A. (2011). Bio-molecular event extraction using support vector machine. In *2011 third international conference on advanced computing* (pp. 298–303).

Sahu, S. K., Christopoulou, F., Miwa, M., & Ananiadou, S. (2019). Inter-sentence relation extraction with document-level graph convolutional neural network. In *Acl.*

Satyapanich, T., Ferraro, F., & Finin, T. (2020a). Casie: Extracting cybersecurity event information from text. *UMBC Faculty Collection*.

Satyapanich, T., Ferraro, F., & Finin, T. (2020b). Casie: Extracting cybersecurity event information from text. In *Proceedings of the association for the advancement of artificial intelligence (aaai).*

Schenk, N., & Chiarcos, C. (2016). Unsupervised learning of prototypical fillers for implicit semantic role labeling. In *Naacl.*

Sevgili, O., Shelmanov, A., Arkhipov, M., Panchenko, A., & Biemann, C. (2020). Neural entity linking: A survey of models based on deep learning. *arXiv preprint arXiv:2006.00575*.

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 *Aaai.*

Shang, Y., Huang, H., Sun, X., & Mao, X. (2020). Learning relation ties with a force-directed graph in distant supervised relation extraction. *arXiv preprint arXiv:2004.10051*.

Shen, Y., Tan, S., Sordoni, A., & Courville, A. (2018). Ordered neurons: Integrating tree structures into recurrent neural networks. *arXiv preprint arXiv:1810.09536*.

Shen, Y., Tan, S., Sordoni, A., & Courville, A. (2019a). Ordered neurons: Integrating tree structures into recurrent neural networks. In *Iclr.*

Shen, Y., Tan, S., Sordoni, A., & Courville, A. (2019b). Ordered neurons: Integrating tree structures into recurrent neural networks. In *Iclr.*

Shi, G., Feng, C., Huang, L., Zhang, B., Ji, H., Liao, L., & Huang, H. (2018). Genre separation network with adversarial training for cross-genre relation extraction. In *Emnlp.*

Silberer, C., & Frank, A. (2012). Casting implicit role linking as an anaphora resolution task. In *The first joint conference on lexical and computational semantics.*

Sims, M., Park, J. H., & Bamman, D. (2019a). Literary event detection. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 3623–3634).

Sims, M., Park, J. H., & Bamman, D. (2019b, 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

Socher, R., Huval, B., Manning, C. D., & Ng, A. Y. (2012). Semantic compositionality through recursive matrix-vector spaces. In *Emnlp.*

Song, L., Zhang, Y., Wang, Z., & Gildea, D. (2018). N-ary relation extraction using graph state lstm. In *Emnlp.*

Song, Y., Wang, J., Jiang, T., Liu, Z., & Rao, Y. (2019). Attentional encoder network for targeted sentiment classification. In *arxiv.*

Soon, W. M., Ng, H. T., & Lim, D. C. Y. (2001). A machine learning approach to coreference resolution of noun phrases. *Computational linguistics*, *27*(4), 521–544.

Souza, F., Nogueira, R., & Lotufo, R. (2020). BERTimbau: pretrained BERT models for Brazilian Portuguese. In *9th brazilian conference on intelligent systems, BRACIS, rio grande do sul, brazil, october 20-23.*

Strubell, E., Verga, P., Andor, D., Weiss, D., & McCallum, A. (2018). Linguistically-informed self-attention for semantic role labeling. In *Emnlp.*

Stylianou, N., & Vlahavas, I. (2019). A neural entity coreference resolution review. *arXiv preprint arXiv:1910.09329*.

Sun, A., Grishman, R., & Sekine, S. (2011). Semi-supervised relation extraction with large-scale word clustering. In *Acl.*

Tai, K. S., Socher, R., & Manning, C. D. (2015). Improved semantic representations from tree-structured long short-term memory networks. In *Acl.*

Thayaparan, M., Valentino, M., Schlegel, V., & Freitas, A. (2019). Identifying supporting facts for multi-hop question answering with document graph networks. In *The thirteenth workshop on graph-based methods for natural language processing at emnlp.*

Tong, M., Wang, S., Cao, Y., Xu, B., Li, J., Hou, L., & Chua, T.-S. (2020). Image enhanced event detection in news articles. In *Proceedings of the association for the advancement of artificial intelligence (aaai).*

Tong, M., Xu, B., Wang, S., Cao, Y., Hou, L., Li, J., & Xie, J. (2020). Improving event detection via open-domain trigger knowledge. In *Proceedings of the annual meeting of the association for computational linguistics (acl).*

Tran, H. M., Nguyen, M. T., & Nguyen, T. H. (2020). The dots have their values: Exploiting the node-edge connections in graph-based neural models for document-level relation extraction. In *Emnlp findings.*

Tran, V.-H., Phi, V.-T., Shindo, H., & Matsumoto, Y. (2019). Relation classification using segment-level attention-based cnn and dependency-based rnn. In *Naacl-hlt.*

Vanegas, J. A., Matos, S., González, F., & Oliveira, J. L. (2015). An overview of biomolecular event extraction from scientific documents. *Computational and mathematical methods in medicine*, *2015*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017a). Attention is all you need. In *Neurnips.*

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017b). Attention is all you need. In *Nips.*

Verga, P., Strubell, E., & McCallum, A. (2018). Simultaneously self-attending to all mentions for full-abstract biological relation extraction. In *Emnlp.*

Verhagen, M., Saurí, R., Caselli, T., & Pustejovsky, J. (2010). SemEval-2010 task 13: TempEval-2. In *Proceedings of the 5th international workshop on semantic evaluation.* Retrieved from `https://aclanthology.org/S10-1010`

Veyseh, A. P. B. (2016). Cross-lingual question answering using common semantic space. In *Proceedings of textgraphs-10: the workshop on graph-based methods for natural language processing* (pp. 15–19).

Veyseh, A. P. B., Dernoncourt, F., Thai, M. T., Dou, D., & Nguyen, T. H. (2020). Multi-view consistency for relation extraction via mutual information and structure prediction. In *Aaai* (pp. 9106–9113).

Veyseh, A. P. B., Nguyen, T. H., & Dou, D. (2019). Improving cross-domain performance for relation extraction via dependency prediction and information flow control. In *Ijcai.*

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

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

Wang, D., Hu, W., Cao, E., & Sun, W. (2020). Global-to-local neural networks for document-level relation extraction. *arXiv preprint arXiv:2009.10359*.

Wang, H., Chen, M., Zhang, H., & Roth, D. (2020a). Joint constrained learning for event-event relation extraction. *arXiv preprint arXiv:2010.06727*.

Wang, H., Chen, M., Zhang, H., & Roth, D. (2020b, 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://www.aclweb.org/anthology/2020.emnlp-main.51`

Wang, L., Cao, Z., de Melo, G., & Liu, Z. (2016). Relation classification via multi-level attention cnns. In *Emnlp.*

Wang, R., Zhou, D., & He, Y. (2019). Open event extraction from online text using a generative adversarial network. *arXiv preprint arXiv:1908.09246*.

Wang, X., Han, X., Liu, Z., Sun, M., & Li, P. (2019a). Adversarial training for weakly supervised event detection. In *Proceedings of the conference of the north american chapter of the association for computational linguistics: Human language technologies (naacl-hlt).*

Wang, X., Han, X., Liu, Z., Sun, M., & Li, P. (2019b). Adversarial training for weakly supervised event detection. 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. 998–1008).

Wang, X., Wang, Z., Han, X., Jiang, W., Han, R., Liu, Z., ... Zhou, J. (2020). MAVEN: A Massive General Domain Event Detection Dataset. In *Proceedings of the conference on empirical methods in natural language processing (emnlp).*

Wang, X., Wang, Z., Han, X., Liu, Z., Li, J., Li, P., ... Ren, X. (2019). Hmeae: Hierarchical modular event argument extraction. In *Emnlp-ijcnlp.*

Wiseman, S. J., Rush, A. M., Shieber, S. M., & Weston, J. (2015). Learning anaphoricity and antecedent ranking features for coreference resolution. 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).*

Wongso, W. (2021). Japanese roberta.. Retrieved from `https://huggingface.co/w11wo/javanese-roberta-small`

Wu, J., Zhang, R., Mao, Y., Guo, H., Soflaei, M., & Huai, J. (2020). Dynamic graph convolutional networks for entity linking. In *Proceedings of the web conference 2020* (pp. 1149–1159).

Wu, L., Petroni, F., Josifoski, M., Riedel, S., & Zettlemoyer, L. (2019). Zero-shot entity linking with dense entity retrieval. *arXiv preprint arXiv:1911.03814*.

Wu, W., Wang, F., Yuan, A., Wu, F., & Li, J. (2020). Corefqa: Coreference resolution as query-based span prediction. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 6953–6963).

Xiang, W., & Wang, B. (2019). A survey of event extraction from text. *IEEE Access*, *7*, 173111–173137.

Xu, Y., Mou, L., Li, G., Chen, Y., Peng, H., & Jin, Z. (2015a). Classifying relations via long short term memory networks along shortest dependency paths. In *Emnlp.*

Xu, Y., Mou, L., Li, G., Chen, Y., Peng, H., & Jin, Z. (2015b). Classifying relations via long short term memory networks along shortest dependency paths. In *Emnlp.*

Yamada, I., & Shindo, H. (2019). Pre-training of deep contextualized embeddings of words and entities for named entity disambiguation. *arXiv preprint arXiv:1909.00426*.

Yan, H., Jin, X., Meng, X., Guo, J., & Cheng, X. (2019a). 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. 5770–5774).

Yan, H., Jin, X., Meng, X., Guo, J., & Cheng, X. (2019b). Event detection with multi-order graph convolution and aggregated attention. In *Proceedings of the conference on empirical methods in natural language processing (emnlp).*

Yang, B., & Mitchell, T. M. (2016a). Joint extraction of events and entities within a document context. In *Naacl-hlt.*

Yang, B., & Mitchell, T. M. (2016b). Joint extraction of events and entities within a document context. In *Proceedings of the conference of the north american chapter of the association for computational linguistics: Human language technologies (naacl-hlt).*

Yang, S., Feng, D., Qiao, L., Kan, Z., & Li, D. (2019). Exploring pre-trained language models for event extraction and generation. In *Acl.*

Yang, Y., Malaviya, C., Fernandez, J., Swayamdipta, S., Le Bras, R., Wang, J.-P., ... Downey, D. (2020). Generative data augmentation for commonsense reasoning. In *Proceedings of the findings of the conference on empirical methods in natural language processing (emnlp).*

Yao, Y., Ye, D., Li, P., Han, X., Lin, Y., Liu, Z., ... Sun, M. (2019). Docred: A large-scale document-level relation extraction dataset. *arXiv preprint arXiv:1906.06127*.

Yu, J., Bohnet, B., & Poesio, M. (2020). Named entity recognition as dependency parsing. *arXiv preprint arXiv:2005.07150*.

Yu, M., Gormley, M. R., & Dredze, M. (2015). Combining word embeddings and feature embeddings for fine-grained relation extraction. In *Naacl-hlt.*

Yuan, Q., Ren, X., He, W., Zhang, C., Geng, X., Huang, L., ... Han, J. (2018). Open-schema event profiling for massive news corpora. In *Proceedings of the conference on information and knowledge management (cikm).*

Yun, S., Jeong, M., Kim, R., Kang, J., & Kim, H. J. (2019). Graph transformer networks. In *Neurips.*

Zelenko, D., Aone, C., & Richardella, A. (2003). Kernel methods for relation extraction. In *Journal of machine learning research.*

Zeng, D., Liu, K., Lai, S., Zhou, G., & Zhao, J. (2014). Relation classification via convolutional deep neural network. In *Coling.*

Zeng, S., Xu, R., Chang, B., & Li, L. (2020). *Double graph based reasoning for document-level relation extraction.*

Zeng, Y., Feng, Y., Ma, R., Wang, Z., Yan, R., Shi, C., & Zhao, D. (2017). Scale up event extraction learning via automatic training data generation. In *Proceedings of the association for the advancement of artificial intelligence (aaai).*

Zhang, D., Li, T., Zhang, H., & Yin, B. (2020). On data augmentation for extreme multi-label classification. *arXiv preprint arXiv:2009.10778*.

Zhang, J., Qin, Y., Zhang, Y., Liu, M., & Ji, D. (2019). Extracting entities and events as a single task using a transition-based neural model. In *Ijcai.*

Zhang, T., Whitehead, S., Zhang, H., Li, H., Ellis, J., Huang, L., ... Chang, S.-F. (2017). Improving event extraction via multimodal integration. In *Proceedings of the 25th acm international conference on multimedia* (pp. 270–278).

Zhang, Y., Qi, P., & Manning, C. D. (2018). Graph convolution over pruned dependency trees improves relation extraction. *arXiv preprint arXiv:1809.10185*.

Zhang, Y., Xu, G., Wang, Y., Lin, D., Li, F., Wu, C., ... Huang, T. (2020). A question answering-based framework for one-step event argument extraction. In *Ieee access, vol 8, 65420-65431.*

Zhang, Y., Yu, X., Cui, Z., Wu, S., Wen, Z., & Wang, L. (2020). Every document owns its structure: Inductive text classification via graph neural networks. *arXiv preprint arXiv:2004.13826*.

Zhang, Y., Zhong, V., Chen, D., Angeli, G., & Manning, C. D. (2017). Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 conference on empirical methods in natural language processing* (pp. 35–45).

Zhang, Z., Kong, X., Liu, Z., Ma, X., & Hovy, E. (2020). A two-step approach for implicit event argument detection. In *Acl.*

Zhou, G., & Su, J. (2002). Named entity recognition using an hmm-based chunk tagger. In *Proceedings of the 40th annual meeting of the association for computational linguistics* (pp. 473–480).

Zhou, G., Su, J., Zhang, J., & Zhang, M. (2005a). Exploring various knowledge in relation extraction. In *Proceedings of the 43rd annual meeting of the association for computational linguistics (acl'05)* (pp. 427–434).

Zhou, G., Su, J., Zhang, J., & Zhang, M. (2005b). Exploring various knowledge in relation extraction. In *Acl.*

Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., & Xu, B. (2016). Attention-based bidirectional long short-term memory networks for relation classification. In *Acl.*

Zwicklbauer, S., Seifert, C., & Granitzer, M. (2016). Robust and collective entity disambiguation through semantic embeddings. In *Proceedings of the 39th international acm sigir conference on research and development in information retrieval* (pp. 425–434).