COLD-START UNIVERSAL INFORMATION EXTRACTION

BY

LIFU HUANG

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2020

Urbana, Illinois

Doctoral Committee:

Professor Heng Ji, Chair
Professor Jiawei Han
Professor Chengxiang Zhai
Associate Professor Kyunghyun Cho

ABSTRACT

Who? What? When? Where? Why? are fundamental questions asked when gathering knowledge about and understanding a concept, topic, or event. The answers to these questions underpin the key information conveyed in the overwhelming majority, if not all, of language-based communication. At the core of my research in Information Extraction (IE) is the desire to endow machines with the ability to automatically extract, assess, and understand text in order to answer these fundamental questions. IE has been serving as one of the most important components for many downstream natural language processing (NLP) tasks, such as knowledge base completion, machine reading comprehension, machine translation and so on. The proliferation of the Web also intensifies the need of dealing with enormous amount of unstructured data from various sources, such as languages, genres and domains.

When building an IE system, the conventional pipeline is to (1) ask expert linguists to rigorously define a target set of knowledge types we wish to extract by examining a large data set, (2) collect resources and human annotations for each type, and (3) design features and train machine learning models to extract knowledge elements. In practice, this process is very expensive as each step involves extensive human effort which is not always available, for example, to specify the knowledge types for a particular scenario, both consumers and expert linguists need to examine a lot of data from that domain and write detailed annotation guidelines for each type. Hand-crafted schemas, which define the types and complex templates of the expected knowledge elements, often provide low coverage and fail to generalize to new domains. For example, none of the traditional event extraction programs, such as ACE (Automatic Content Extraction) and TAC-KBP, include "donation" and "evacuation" in their schemas in spite of their potential relevance to natural disaster management users. Additionally, these approaches are highly dependent on linguistic resources and human labeled data tuned to pre-defined types, so they suffer from poor scalability and portability when moving to a new language, domain, or genre.

The focus of this thesis is to develop effective theories and algorithms for IE which not only yield satisfactory **quality** by incorporating prior linguistic and semantic knowledge, but also greater **portability** and **scalability** by moving away from the high cost and narrow focus of large-scale manual annotation. This thesis opens up a new research direction called **Cold-Start Universal Information Extraction**, where the full extraction and analysis starts from scratch and requires little or no prior manual annotation or pre-defined type

schema. In addition to this new research paradigm, we also contribute effective algorithms and models towards resolving the following three challenges:

- How can machines extract knowledge without any pre-defined types or any human annotated data? We develop an effective bottom-up and unsupervised *Liberal Information Extraction* framework based on the hypothesis that the meaning and underlying knowledge conveyed by linguistic expressions is usually embodied by their usages in language, which makes it possible to automatically induces a type schema based on rich contextual representations of all knowledge elements by combining their symbolic and distributional semantics using unsupervised hierarchical clustering.

- How can machines benefit from available resources, e.g., large-scale ontologies or existing human annotations? My research has shown that pre-defined types can also be encoded by rich contextual or structured representations, through which knowledge elements can be mapped to their appropriate types. Therefore, we design a weakly supervised *Zero-shot Learning* and a *Semi-Supervised Vector Quantized Variational Auto-Encoder* approach that frames IE as a grounding problem instead of classification, where knowledge elements are grounded into any types from an extensible and large-scale target ontology or induced from the corpora, with available annotations for a few types.

- How can IE approaches be extent to low-resource languages without any extra human effort? There are more than 6000 living languages in the real world while public gold-standard annotations are only available for a few dominant languages. To facilitate the adaptation of these IE frameworks to other languages, especially low resource languages, a *Multilingual Common Semantic Space* is further proposed to serve as a bridge for transferring existing resources and annotated data from dominant languages to more than 300 low resource languages. Moreover, a *Multi-Level Adversarial Transfer* framework is also designed to learn language-agnostic features across various languages.

To my parents, for their love and support.

ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest gratitude to my advisor, Professor Heng Ji, for her continuous and tremendous support, encouragement, and advising. I have always been feeling fortunate and honored of being one of her students. I will never forget the call that I received from her, telling me that she was working hard to get me admitted to the Ph.D. program. Over the past six years, Heng has provided me a lot of valuable suggestions and advice to improve my skills in writing, communication, presentation as well as all other aspects in terms of doing good research. Heng has been very open-minded and extremely supportive, and has provided me much freedom to explore my own research interests and topics. She treats each student as a peer, a friend and a colleague. I will never forget the moments that we had in Winslow building, working hard together and deep into night on paper submissions and project evaluations. During my last year at University of Illinois at Urbana-Champaign (UIUC), Heng also spent a lot of effort on helping me improve the application materials, practice the job talk and interviews. Most importantly, I especially admire Heng's high enthusiasm and passion for scientific research, and her diligent work attitude, which will guide me in my future career. No matter how much I write here, I cannot express my gratitude to her.

Besides my advisor, I would like to extend my gratitude to the rest of my thesis committee: Professor Jiawei Han, Professor Chengxiang Zhai and Professor Kyunghyun Cho. It is my great honor to have them on my committee. Their insightful comments and questions have been extremely helpful to improve this thesis. Professor Jiawei Han and Professor Kyunghyun Cho have also kindly provided me a lot of valuable advice and guidance during the past research collaborations and discussions. I benefit a lot from their humble and rigorous research attitude and extensive knowledge, from which I learned how to be a respectable researcher.

My sincere thanks also goes to all the members of Blender lab, Dr. Tongtao Zhang, Dr. Boliang Zhang, Dr. Dian Yu, Dr. Di Lu, Dr. Hao Li, Xiaoman Pan, Ying Lin, Spencer Whitehead, Manling Li, Qingyun Wang, Qi Zeng, Pengfei Yu, Haoyang Wen, for the insightful discussions and wonderful collaborations on research projects. Each of them has some very special and outstanding characters, from which I benefited a lot. I also feel very lucky to have the chance to work and collaborate with other visiting students and academia friends, Dr. Tao Ge, Dr. Xiaocheng Feng, Dr. Yixin Cao, Dr. Ge Shi, Dr. Wenpeng Yin, from whom I learned tremendous knowledge and research skills.

I'm also very grateful to all my mentors, Dr. Taylor Cassidy, Dr. Clare Voss, Dr. Chin-Yew Liu, Dr. Jing Liu, Dr. Avirup Sil, Dr. Radu Florian, Dr. Peter Clark, Dr. Scott Wen-tau Yih, Dr. Bhavana Dalvi, Dr. Niket Tandon, Dr. Yejin Choi, Dr. Ronan Le Bras, Dr. Chandra Bhagavatula, for their mentorship and guidance during the past projects and internships. These experiences made my life more enjoyable and extended my research to broader areas.

Last but not the least, I would like to thank my parents, grandparents and my sister for their tremendous patience and support for me to pursue Ph.D. degree as well as the support throughout my whole life.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

The majority of the current information sources are based on natural language text, ranging from books, news articles, social media posts, scientific journals, to a wide range of textual information from various domains and languages, such as electronic medical records, financial or government reports. Understanding *who did what to whom, when and where* from such a massive unstructured text corpora is quite painstaking as human needs to read and understand a huge amount of information. Structured data, such as a graph consisting of entities, events and their relationships, is serving as an important source for human to understand the world. It allows people to quickly understand the text and retrieve the information that they need in an efficient and unambiguous manner. The bulk of this thesis is devoted to the problem of automatically turning unstructured text into structured knowledge.

Information Extraction (IE) is the task of automatically extracting concepts (e.g., entities and events) and their relations from unstructured texts. It has been a popular research topic in natural language processing (NLP) since 1990s, when the series of Message Understanding Conferences (MUCs) [1] were introduced. The input to an IE system is a set of texts, e.g., news-wire articles, tweets, and the output is a set of structured and unstructured facts. According to the types of these facts, IE can be divided into multiple downstream subtasks, e.g., named entity recognition, relation extraction which determines the relationship between two or more entities, and event extraction that detects event triggers as well as their participants with particular roles. To help researchers better understand the task and algorithms, we provide a detailed definition for each term involved all the sub-tasks of IE.

- An **Entity** is an object or set of objects in the real world and a **Mention** is a reference to a particular entity. Entities can be referenced in a text by their name or pronoun.[1]
- A **Relation** defines a semantic relationship between two entities. Relations are usually characterized based on orderd pairs of entities.[2]
- An **Event** is a specific occurrence involving participants, which is frequently described as a change of state. An **Event Trigger** is the word that most clearly expresses the occurrence of an event, and an **Event Argument** is a concept that serves as a participant in an event. Each argument also plays an **Argument Role**, which specifies the function or purpose of an argument.[3]

[1] https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/english-entities-guidelines-v6.6.pdf
[2] https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/english-relations-guidelines-v5.8.3.pdf
[3] https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/english-events-guidelines-v5.4.3.pdf

Figure 1.1 shows three example sentences and their IE outputs. For example, in E3, *troops* is recognized as a mention of a Person entity and *Himalayan* is identified as a mention of a Location entity. They hold a semantic relationship named *Located*. In addition, a Transport-Person event mention is also detected which is triggered by *dispatching* and has multiple arguments involved, e.g., the Government of China is the Agent argument, troops is a Person argument, Himalayan is the Destination and 1950 is the Time.

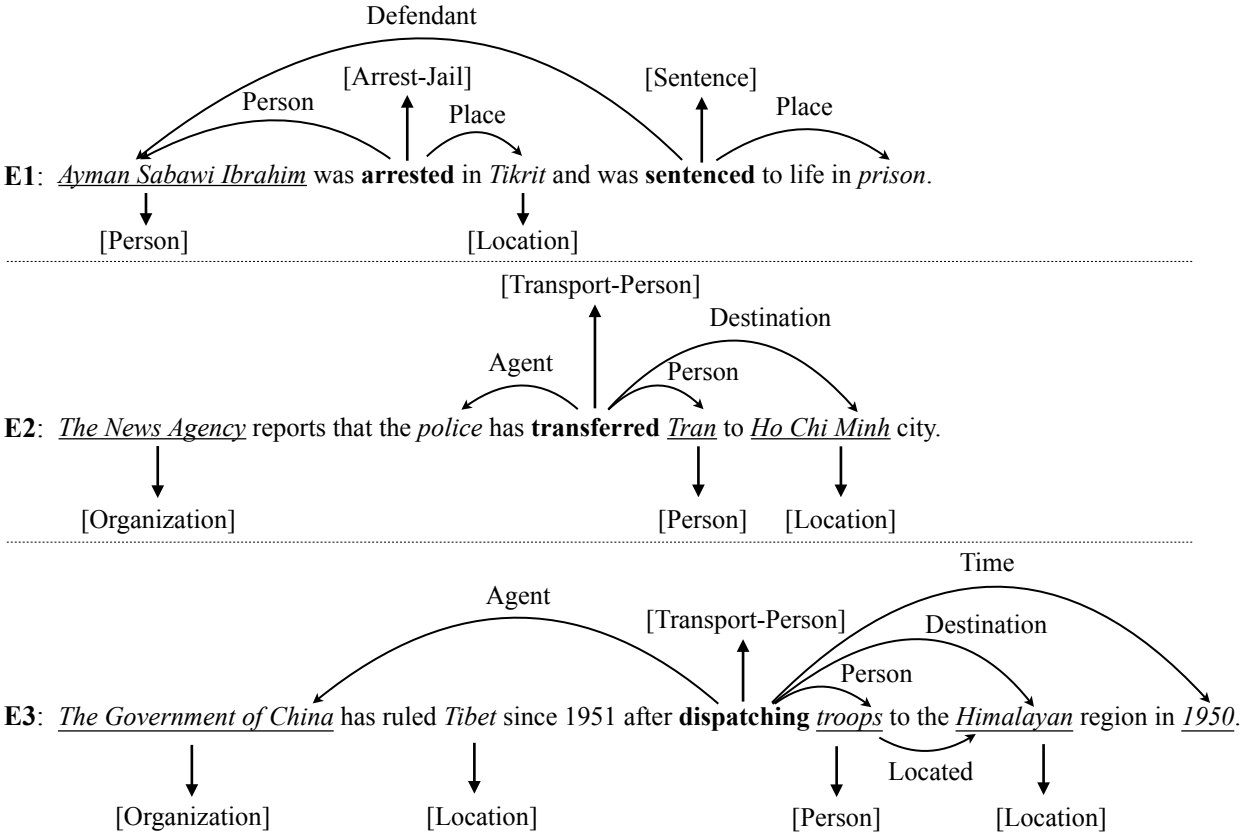

Figure 1.1: Example of Information Extraction Output. The upper annotations show event extraction output while the lower annotations show entity and relation extraction output.

Given a natural language sentence, an IE system should be able to identify multi-level structured information. For example, for sentence *E3*, it should first recognize what types of names are included, e.g., the organization name *The Government of China*, the location name *Tibet*, or the nominal mention *troops*, and then determine the relationship between each pair of names, e.g., the *Located* relationship between *troops* and *Himalayan*. In addition, an IE system should also detect *dispatching* as a trigger for an *Transport-Person* event mention, and identify *Himalayan* as the *Destination* and *1950* as the *Time* for *dispatching* event.

Recently, as the development of machine learning technologies, IE systems have been applied to a wide range of textual sources: from high resource languages (e.g., English,

Chinese) [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14] to extremely low resource languages (e.g., Amharic, Uyghur) [15, 16, 17, 18, 19, 20], from general news domain to biomedical domain [21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31] or scientific domain [32, 33], and even from plain text to multimedia (e.g., images, videos) [34, 35].

1.1 TRADITIONAL IE PIPELINES AND THEIR LIMITATIONS

With improving the quality of extraction as the main goal, traditional IE has gone through a "hill-climbing" process since 1990s, driven by various shared tasks including MUC [1], CONLL [36, 37], ACE and TAC-KBP [38, 39]. This paradigm includes three steps:

1. Some expert linguists define a schema about "what to extract", such as concept types and their relations for a specific data collection based on the needs of potential users, and write an annotation guideline for each type in the schema;

2. Human annotators follow the guideline to annotate a certain amount of documents (a typical size is 500 documents);

3. Researchers design features and train supervised learning models from these manually annotated data.

This traditional IE paradigm has achieved significant successes on some IE tasks. For example, for English name tagging task, current state-of-the-art approaches [16, 40, 41, 42] can achieve more than 90% F-score on CONLL2003 [36, 37] benchmark using 15,000 annotated sentences with more than 23,000 annotated names, while for English event extraction task, state-of-the-art approaches [13, 35, 43] can achieve more than 70% F-score for event trigger extraction and 58% F-score for argument extraction with 5000 annotated event mentions. However, we argue that it has at least two limitations which make it difficult to be directly applied to new domains or languages:

1. First, this paradigm is not fully automatic because it involves human in the loop during the first two steps. Both of the predefined type schema and human annotated data are very expensive. For example, it took expert linguists almost one year to define the ACE (Automatic Content Extraction) event type schema which covers 33 event types, and government spent millions of dollars to hire human annotators to fully annotate 599 articles.

2. Second, a predefined schema can only cover a limited number of types and relations. For example, some typical types of emergent events such as "donation" and "evacuation" in response to a natural disaster are missing in all of the traditional IE programs. Extracting event mentions for new types means to start the whole process again from zero, without utilizing any resources for existing types.

Some recent research efforts have been made to address these limitations. Several new paradigms such as Open IE [44, 45, 46], Preemptive IE [47] and On-Demand IE [48] exploited data-driven methods (e.g., domain-independent patterns) to extract relations or events which are not restricted to any pre-defined schema. However, state-of-the-art Open IE techniques obtained substantially lower recall than traditional IE [46], because they heavily rely on information redundancy and thus fail to discover relations in the "long-tail" due to knowledge sparsity. In addition, they are still incapable of generalizing contexts to name new relation types and form a schema.

1.2 MOTIVATIONS AND SOLUTIONS

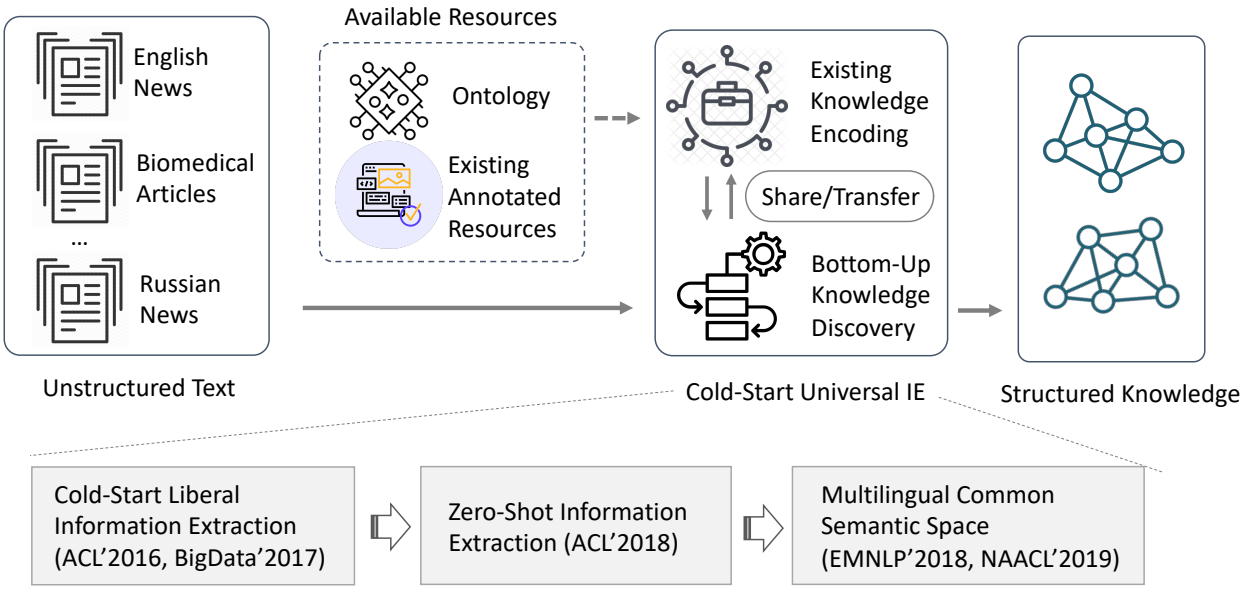

Figure 1.2: Overview of Our Cold-Start Universal Information Extraction Paradigm.

To solve these limitations, we basically need to answer three research questions: (1) in order to reduce the human cost, how can machines automatically discover the key information without any human effort? (2) to improve the coverage, can machines automatically discover new types of knowledge by leveraging existing annotations? (3) to improve the portability,

we also need to investigate how can the knowledge be effectively shared and transferred across various scenarios, e.g., from one domain to another domain, or from one language to another language.

This thesis addresses these questions with a new research paradigm, Cold Start Universal Information Extraction, a "bottom-up" information extraction fashion which attempts to discover structured knowledge from any source of unstructured text, such as English news articles, Russian news articles or even Biomedical articles, by leveraging all available resources, including manually defined ontologies or existing annotated data for old scenarios without requiring human in the loop. As Figure 1.2 shows, the universal information paradigm covers three representative work: a cold-start liberal IE approach, a zero-shot transfer learning framework, as well as a multilingual common semantic space, which serves as a bridge for knowledge transfer across various domains, sources, and languages.

1.2.1 Liberal Information Extraction

Traditional IE approaches tend to follow a "top-down" manner - learning effective features for each predefined type according to human annotated data, and then discovering the facts specific to the predefined types. We take a fresh look at the IE problem and design a cold-start liberal information extraction framework. We argue that all the types of candidate facts can be discovered with simple features or existing linguistic resources. For example, for entity recognition, a lot of work [49, 50, 51, 52] has explored to apply patterns and lexical features to identify names; which for event extraction, we can take all the verbs as well as the nouns that are covered by exiting linguistic resources, e.g., FrameNet [53], VerbNet [54], propbank [55], and ontonotes [56], as candidate triggers, and take all the semantically related contexts in a semantic parsing output, e.g., Abstract Meaning Representation (AMR) parsing or dependency parsing output, as their candidate arguments. This hypothesis can be verified by the fact that about 90% of the triggers in ACE2005 annotation can be covered by the candidate triggers discovered by AMR parsing and FrameNet. After identifying all candidate facts, the next step is to automatically induce a type schema. We observed that the facts, such as entities and event triggers, usually shared similar types when they occur in similar contexts and scenarios. Thus, in order to automatically induce a type schema, we seek to group all the candidate facts into various clusters based on their semantics and each cluster denotes one type. Previous work [12, 13, 57] usually take the contextual words and distributional semantic representations as features to a classifier and map each trigger to a predefined type. While in Liberal IE framework, we use such features to represent the rich semantics of each fact and specify its type. The facts which share similar semantics as well

as local contexts should belong to similar types. For example, in Figure 1.1, the semantics of the *dispatching* event mention in $E3$ should be closer to the semantics of *transferred* event mention in $E2$ than that of *arrested* or *sentenced* event mention in $E1$. Without requiring any annotated data or predefined type schema, this framework can be directly applied to any genres, domains, or languages, and can automatically induce a type schema which is customized to the input corpus. Taking event extraction task as an example, Figure 1.3 depicts the differences between traditional ACE paradigm and our liberal IE paradigm.

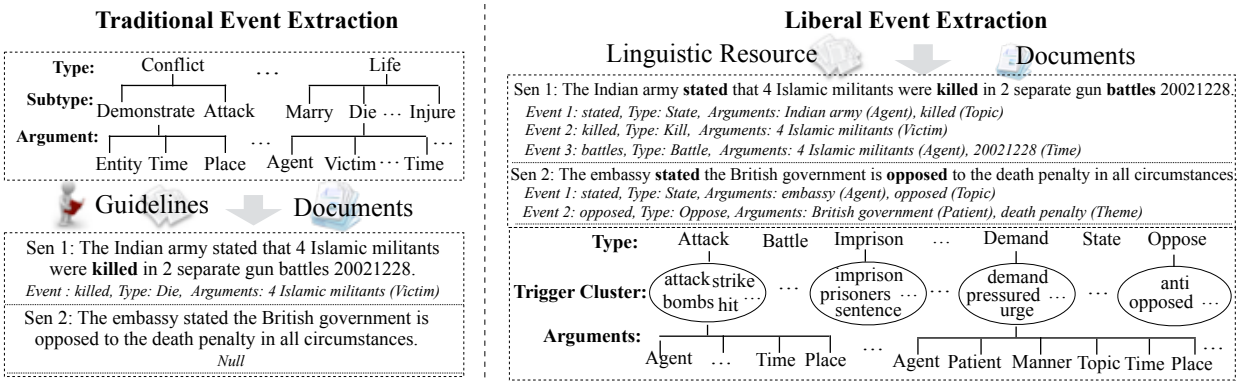

Figure 1.3: Comparison between ACE Event Extraction and Liberal Event Extraction.

1.2.2 Zero-Shot Information Extraction

Liberal information extraction approach can bottom up discover facts and automatically induce a type schema from the given input corpus. However, in many practical applications, the systems are required to extract certain predefined types of facts without providing any in-domain annotated data. In this case, the types that are automatically discovered by liberal IE approach may not be able to automatically or perfectly mapped with the given types. On the other hand, expert linguists have spent decades on creating ontologies and linguistic resource, and governments have spent millions of dollars on collecting annotated data for the existing type schemas. For example, several event extraction programs have been proposed in recent years. ACE (Automatic Content Extraction) program defines 8 main event types and 33 event subtypes across three languages, English, Arabic and Chinese, while ERE (Entities, Relations, Events) consists of 38 event subtypes. Both of these two programs have released thousands of annotated articles according to the guidelines written by expert linguists.

Considering these facts, we go beyond the liberal IE approach, and rise a new research question: can we take advantage of the existing ontologies, as well as the annotations for

existing types, to discover new types of facts? We finally achieve this goal by adopting a zero-shot transfer learning framework for IE. Similar as the liberal IE approach, we first identify all the candidate facts and get their rich semantic representations by incorporating their local contexts. Then, inspired by the theory of [58] for event extraction task: "the semantics of an event structure can be generalized and mapped to event mention structures in a systematic and predictable way", we propose to model information extraction as a grounding problems, instead of a classification or clustering problem, by learning a regression function to map each fact to its semantically closest type. The mapping function is independent of types, and can be trained from annotations for a limited number of seen types and further used for any new unseen types. With this zero-shot learning framework, we can take advantage of all available linguistic resources as well as annotations for existing seen types, and automatically discover facts for any new given types without requiring any human effort.

To alleviate the reliance of the large-scale target ontology, which is usually unavailable for many scenarios, we further extend the zero-shot IE framework to a more challenging setting, where the approaches are required to extract knowledge elements and induce a set of new event types. We design a semi-supervised vector quantized variational auto-encoder approach to automatically learn a vector representation for each potential unseen type and ground each candidate fact into a seen or unseen type. With event detection as a case study, this approach is approved to be able to discover a set of high-quality unseen types given the annotations for a few seen types.

1.2.3 Multilingual Common Semantic Space

Both liberal information extraction and zero-shot learning frameworks are built upon the rich semantic representations of each candidate fact. In order to facilitate the adaptation of these approaches to new languages, we further propose to construct a multilingual common semantic space, where words from multiple languages are projected into a shared semantic space to serve as the bridge for knowledge transfer.

Previous work usually learn multilingual word embeddings using bilingual word dictionaries, which, however, are not always enough, especially for low resource languages. Though several recent attempts [59, 60, 61] have shown that it is possible to extract multilingual word embedding from a pair of potentially unaligned corpora in multiple languages, we claim that it is necessary to impose more constraints to preserve linguistic properties and facilitate downstream NLP tasks, such as cross-lingual IE, and MT. We find that words also can be clustered through explicit (e.g., sharing affixes of certain linguistic functions) or implicit clues (e.g., sharing neighbors from monolingual word embedding) and such clusters

should also be consistent across multiple languages. To do so, we introduce multiple cluster-level alignments and enforce the word clusters to be consistently distributed across multiple languages. We exploit three signals for clustering: (1) neighbor words in the monolingual word embedding space; (2) character-level information; and (3) linguistic properties (e.g., apposition, locative suffix) derived from linguistic structure knowledge bases available for thousands of languages. We introduce a new cluster-consistent correlational neural network to construct the common semantic space by aligning words as well as clusters. By encouraging the consistency of clusters, this framework can maintain high performance with very small size of bilingual lexicons, thus can serve as a bridge for transferring linguistic resources as well as existing human annotations across thousands of languages.

In addition, we also explore adversarial training for cross-lingual transfer and design a multi-level adversarial training framework to efficiently learn language-agnostic features on both word level and sequence level. Using low-resource name tagging task as a case study, it achieves up to 16% absolute F-score gain overall high-performing baselines on cross-lingual transfer without using any target-language resources.

1.3 NOVELTY AND CONTRIBUTION CLAIMS

The main contribution of this thesis is to investigate the main limitations of traditional approaches for information extraction task and propose several new architectures to solve the problems. Here we briefly summarize the main contributions of the three representative frameworks:

- To the best of our knowledge, Liberal IE is the first information extraction paradigm to take human out of the IE loop and bring IE systems into the joy of identifying useful information liberally. A Liberal IE approach can simultaneously discover a domain-rich schema which is customized to the input data, and extract structured knowledge. It has an absolute "cold-start" and can be adapted to any genres, domains, and languages without any human annotated data, and thus tremendously save human cost. The resulting system is being successfully used by various government agencies (e.g., ARL, ARFL, and IARPA) and industrial companies (e.g., Bosch, IBM) on various domains (e.g., military, disaster, bio-medical, power tool). It has also been widely cited and has inspired follow-up research on open-domain information extraction, event representation learning, event-event relation prediction.

- We also propose a new view of information extraction and reframe it from the current classification into a grounding problem, and design a zero-shot information extraction

framework and a semi-supervised vector quantized variational auto-encoder approach. Taking advantage of the annotations for very limited types of old scenarios, our zero-shot IE approach can achieve comparable performance for thousands of types as supervised methods without requiring extra human annotation effort. As a result of these efforts, the extraction capabilities have been extended from dozens of types (e.g., 33 types for event extraction) to more than 1000 types while ensuring high quality.

- We also introduce an elegant way of effectively transferring available resources (e.g., manually constructed ontology or manually annotated data) across various languages. A multilingual common semantic space is constructed to serve as a bridge among thousands of languages. Without requiring large size of bilingual dictionaries and multilingual descriptions for the same image, we take advantage of multi-level cluster alignments between each pair of languages, and can automatically align multilingual words in a shared semantic space. The resulting embeddings better retain the clustering structures in each language, which is important to multi-lingual IE. This work enables IE to be feasible for thousands of languages without requiring any human effort. By leveraging available resources from English through the common semantic space, we provide coordinated NER (Named Entity Recognition) for hundreds of languages (e.g., Turkish, Amharic, Uyghur) without parallel data and achieve up to 24.5% absolute F-score gain.

1.4 THESIS STRUCTURE

The rest of this thesis is structured as follows:

- Chapter 2 provides a comprehensive literature overview for previous information extraction studies, including traditional rule-based and supervised IE approaches, semi-supervised and distantly supervised methods, and the line of open information extraction research. It also compares all these previous studies with our new cold-start universal IE approaches in terms of the requirement of human effort and the coverage and portability of the methods. : describes previous studies that related to our topics.

- Chapter 3 describes the main framework of our liberal information extraction approach. We demonstrated the quality and portability of this framework on both event extraction and fine-grained named entity typing tasks.

- Chapter 4 presents idea of zero-shot learning for event extraction task. The system trained on 500 sentences for 6 types and tested on 27 new types achieves comparable

performance as a supervised extractor trained on more than 3000 sentences for 33 types.

- Chapter 5 discusses an extension of the zero-shot learning framework, a semi-supervised vector quantized variational autoencoder approach, which does not required a target ontology, and instead, can automatically induce a set of new unseen types based on available annotations for a few seen types.

- Chapter 6 investigates the problems of extending IE from dominant languages to low-resource languages by constructing a multilingual common semantic space. Intrinsic evaluation on monolingual and multilingual QVEC tasks and extrinsic evaluation on low-resource name tagging task demonstrates the effectiveness of this common space.

- Chapter 7 discusses a cross-lingual adversarial transfer approach which can be applied to the scenario where no bilingual dictionaries or alignments are available. By leveraging word-level and sequence-level adversarial training, this approach achieves up to 16% F-score gain on low-resource name tagging task.

- Finally, Chapter 8 presents the conclusion and contributions of this thesis. We also discuss the remaining limitations of current frameworks and algorithms, and point out to some directions for future research.

CHAPTER 2: RELATED WORK

This chapter first provides a comprehensive review for the traditional information extraction approaches as well as the methods that are based on other information extraction paradigms, and then compares our new Cold-Start University IE approaches with these prior arts in terms of the quality as well as the requirement of human effort.

2.1 TRADITIONAL INFORMATION EXTRACTION

2.1.1 Rule and Pattern Based Approaches

Information extraction has long been an active research topic in natural language processing. The earliest stage of IE approaches that are designed to detect entities, relations and events are based on human-crafted heuristic rules or patterns [62, 63, 64, 65, 66, 67, 68, 69, 70]. For example, to identify Person entity names, [71] summarized more than 100 rules and patterns as Table 2.1 shows. In addition, [62] described a TextMarker system to acquire and refine a set of rules for structured data extraction. [64] proposed a generative model that incorporates distributional prior knowledge to help distribute candidate slot fillers in a document into appropriate slots and identify meaningful template slots. In addition, many rules, such as dependency restrictions [72], entity type constraints [73], seed dictionaries [74], are also combined with supervised learning approaches.

<Token> ["$[A-Z][a-z]*$"]	→	<CapsWord>
<Token> ["Michael | Richard | Smith |..."]	→	<PersonDict>
<Token> ["Mr. | Mrs. | Dr. | Miss. |..."]	→	<Salutation>
<PersonDict> <PersonDict>	→	<Person>
<Salutation> <CapsWord>	→	<Person>
<Salutation> <CapsWord> <CapsWord>	→	<Person>
... ...	→	... ...

Table 2.1: Rules and Patterns for Identifying Person Entity Names.

With hand crafted rules and patterns, these IE approaches tend to generate very fast and good results with very high precision. The identification process is easy to comprehend and trace as we can quickly tell based on which rules is the entity or relationship be determined, so that developers can quickly fix the cause of errors. Another advantage is that domain specific knowledge can be easily incorporated into the rule based IE systems as additional

dictionary or patterns. Given these pros, many commercial vendors are relying on rule-based IE systems. However, it's also noticeable that these approaches also require tedious manual labor from domain experts to identify and correctly define the rules and patterns. According to the analysis on the EMNLP, ACL, and NAACL conference proceedings from 2003 through 2012 [75], only 3.5% of the research papers on information extraction are purely based on rules and patterns.

2.1.2 Supervised Machine Learning Based Approaches

Most of the current information extraction methods were based on human annotated data and supervised learning techniques, which can be further divided into: (1) machine learning models using manually crafted features, and (2) deep neural networks with distributional semantic embedding features. This section briefly summarizes the supervised algorithms and models for each subtask of IE.

Name tagging task is usually treated as a sequence labeling problem, where each token is mapped to a tag based on its feature representation. Early machine learning models that have been explored include Hidden Markov Model (HMM) [76, 77, 78, 79], Support Vector Machines (SVMs) [80], Conditional Random Fields (CRFs) [81, 82], and decision trees [83]. As each name may contain multiple tokens, CRFs has been proved to be quite effective in capturing inter dependency between name tags [84, 85]. [86] is the first to use neural network architecture for name tagging task, where feature vectors are constructed from orthographic features and a multi-layer feed forward neural networks is used for label prediction. This approach was further improved by replacing the manually constructed features with word embeddings [87]. Afterwards, Recurrent Neural Networks, especially Bi-directional Long Short-Term Memory (Bi-LSTM) networks [41, 88, 89, 90], showed significant improvements on all the sequence labeling tasks. They process each sequence in both directions with two separate hidden layers to learn a contextual representation for each token, which is then fed into a CRFs layer to predict a name tag.

Event extraction task aims to extract both a trigger and its arguments, where the argument candidates are the entities detected from the same sentence. Early research on event extraction mainly relies on local sentence-level symbolic features in a pipelined architecture [12, 57, 91, 92, 93, 94, 95], where the extraction of triggers, which tags each token to a particular event type or *Other*, and argument links, which are classified for each pair of a candidate trigger and an argument into a particular argument role or *Other*, are modeled as two isolated subtasks. Such pipelined approaches prohibit the interaction among components such that errors from upstream components are propagated to downstream ones.

To resolve this limitation, joint models using Markov Logic Network [96], Structured Perceptrons [97, 98, 99], Dual Decomposition [29] and Deep Neural Network [100, 101] are developed to jointly identify the candidate trigger and arguments by considering the interactions between these two predictions.

Both supervised learning and rule based IE approaches usually achieve high quality for known types, but cannot be directly applied to any new types. Both of them require human to be involved: experts need to define a type schema with detailed guidelines; system developers either manually defined a set of hand-written rules or ask human annotators to annotate a set of articles according to the guidelines. Handling new types means to start over from zero and repeat the same effort. In contrast, our liberal IE framework leverages rich semantic representations of knowledge elements and requires no human annotations or human predefined types. However the performance is not as good as supervised approaches. These two lines of approaches could possibly be combined and complementary to each other, for example, the new type distributions obtained from liberal IE can be taken as high-level effective features to supervised approaches.

Another limitation of these supervised learning approaches is that they usually model the target types as atomic symbols, thus they can only measure the similarity between features encoded for testing data and annotated data. However, our zero-shot IE framework further incorporates the semantics of the types into the learning process, thus it can be applied to any new types because the semantics of the types is independent of the annotated data. Similar ideas haven been incorporated into supervised learning approaches to improve the extraction performance [102, 103, 104].

2.2 OPEN INFORMATION EXTRACTION

Open information extraction (OIE) [45, 105, 106, 107, 108, 109] is a recently proposed extraction paradigm that facilitates domain independent discovery of relations or textual assertions, consisting in a verb relation and two arguments, extracted from text and scales to the diversity and size of the Web corpus. Unlike most traditional information extraction methods which focus on a limited set of predefined relation types, an open IE system can extract any types of verbal relations found in the text. Most of previous open IE systems explore lexical or syntactic features and patterns to extract relational triples within a sentence. Among which, TextRunner [45] is based on a second order linear-chain CRF trained on triples sampled from Penn Treebank while the input features include part-or-speech tags and NP-chunked sentences. WOE [106] extract triples by identifying the shortest dependency paths between two noun phrases. The state-of-the-art OIE system, ReVerb [110], takes the

same set of syntactic features as TextRunner as input to a logic regression classifier, and incorporates lexical constraints to filter out over-specified relational triples.

Recently, deep neural networks (DNNs) have also been explored for open information extraction. [111, 112, 113] explored recurrent neural networks or neural encoder-decoder framework with attention and copying mechanisms to automatically discover tuples from each natural language sentence. [114] further proposes a novel supervised open information extraction framework that leverages an ensemble of unsupervised Open IE systems and a small amount of labeled data to improve system performance. To discover implicit relational tuples, [115, 116] also explored reading comprehension datasets for Open IE. Several studies [117, 118, 119] also extend Open IE to cross-lingual and multi-lingual by leveraging sequence-to-sequence translation models or multilingual embeddings.

Both open IE and our Liberal IE approaches leverage syntactic or semantic parsing outputs to discover the relations or events, which are not restricted to any pre-defined schema and dramatically enhance the scalability of IE. However, the outputs of open IE approaches are usually a massive amount of tuples, which are difficult to discover the connections or underlying relations among tuples, while our liberal IE approach further induces a type schema, a typological representation of the events or entities, by combining symbolic and distributional semantics of the knowledge elements. Our zero-shot IE approach can further ground all candidate knowledge elements to a large-scale and extensible target ontology, thus it can be combined with open IE approaches, e.g., directly mapping the relation or event tuples discovered by open IE approaches to a target ontology.

2.3 WEAKLY AND DISTANTLY SUPERVISED INFORMATION EXTRACTION

Recently, distant supervision has been widely applied in information extraction tasks, especially on relation extraction. [120, 121] use weakly labeled data in bioinformatics for biological knowledge base construction. [122] first exploited WordNet for extracting hypernym(is-a) relations between entities. In addition, many previously studies [7, 123, 124, 125, 126, 127, 128, 129, 130] use the entity pairs extracted from existing knowledge base, e.g. Wikipedia, DBPedia, Freebase, as signals to acquire weakly annotated data for relation extraction.

Though, distant supervision can help reduce human annotation effort, it still requires human effort to map the types from knowledge based to the target type schema. More importantly, the weakly supervised data extracted with distant supervision are usually noisy, which hinders the performances in real applications. Figure 2.1 compares our new cold-start universal IE paradigm with traditional rule-based and supervised IE approaches, as well as distantly supervised IE approaches in terms of the requirement of human effort and coverage

and portability of the approaches.

Figure 2.1: Comparison Among Various Information Extraction Paradigms.

CHAPTER 3: COLD-START LIBERAL INFORMATION EXTRACTION

In this chapter, we present a liberal information information framework which can bottom-up discover candidate facts from a given corpora and automatically induce a type schema which is customized to it. We use event extraction as a case study to describe how this framework works and demonstrate the effectiveness of this framework on both general news domain and biomedical domain.

3.1 MOTIVATIONS

The main research problem that we are addressing here includes three subtasks: (1) how to automatically discover all candidate facts from unstructured texts? (2) how to get a rich semantic representation for each candidate fact and (3) how to induce a type schema and automatically assign a type for each fact. For the first subtask, we observe that the candidate facts, such as event triggers and arguments, can be automatically discovered based on some simple patterns, lexical features and existing linguistic resources. For example, most of the verbs can be regarded as candidate triggers and are covered by existing linguistic resources, e.g., FrameNet [53], VerbNet [54], propbank [55], and ontonotes [56], and most of the noun phrases can be regarded as candidate entity mentions.

The second subtask of learning rich semantic representation for each candidate fact is the core of our liberal IE framework. The semantic representation should be able to capture local contexts and specify its type. Let's consider the following examples to describe our motivations:

E1. Two Soldiers were **killed** and one **injured** in the close-quarters **fighting** in Kut.

E2. *Bill Bennet*'s glam *gambling* **loss** changed my opinion.

E3. Gen. Vincent Brooks announced the **capture** of *Barzan Ibrahim Hasan al-Tikriti*, telling reporters he was an adviser to Saddam.

E4. This was the *Italian ship* that was **captured** by *Palestinian* terrorists back in 1985.

E5. *Ayman Sabawi Ibrahim* was **arrested** in *Tikrit* and was **sentenced** to life in *prison*.

We seek to cluster the event triggers and event arguments so that each cluster represents a type. We rely on distributional similarity for our clustering distance metric. The distributional hypothesis [131] states that words often occurring in similar contexts tend to

have similar meanings. We formulate the following distributional hypotheses specifically for information extraction, and develop our approach accordingly.

Hypothesis 3.1: *Event triggers that occur in similar contexts and share the same sense tend to have similar types.*

Following the distributional hypothesis, when we simply learn general word embeddings from a large corpus for each word, we obtain similar words like those shown in Table 3.1. We can see similar words, such as those centered around "*injure*" and "*fight*", are converging to similar types. However, for words with multiple senses such as "*fire*" (shooting or employment termination), similar words may indicate multiple event types. Thus, we propose to apply Word Sense Disambiguation (WSD) and learn a distinct embedding for each sense.

injure	Score	fight	Score	fire	Score
injures	0.602	fighting	0.792	fires	0.686
hurt	0.593	fights	0.762	aim	0.683
harm	0.592	battle	0.702	enemy	0.601
maim	0.571	fought	0.636	grenades	0.597
injuring	0.561	Fight	0.610	bombs	0.585
endanger	0.543	battles	0.590	blast	0.566
dislocate	0.529	Fighting	0.588	burning	0.562
kill	0.527	bout	0.570	smoke	0.558

Table 3.1: Top-8 Most Similar Words (in 3 Clusters)

Hypothesis 3.2: *Beyond the lexical semantics of a particular event trigger, its type is also dependent on its arguments and their roles, as well as other words contextually connected to the trigger.*

For example, in *E4*, the fact that the patient role is a vehicle ("*Italian ship*"), and not a person (as in *E3* and *E5*), suggests that the event trigger "*captured*" has type "*Transfer-Ownership*" as opposed to "*Arrest*". In *E2*, we know the "*loss*" event occurs in a *gambling* scenario, so we can determine its type as loss of money, not loss of life.

We therefore propose to enrich each trigger's representation by incorporating the distributional representations of various words in the trigger's context. Not all context words are relevant to event trigger type prediction, while those that are vary in their predictive value. We propose to use semantic relations, derived from a meaning representation for the text, to carefully select arguments and other words in an event trigger's context. These words are then incorporated into a "global" event structure for a trigger mention. We rely on semantic relations to (1) specify how the distributional semantics of relevant context words contribute to the overall event structure representation; (2) determine the order in which distributional

semantics of relevant context words are incorporated into the event structure.

3.2 APPROACH OVERVIEW

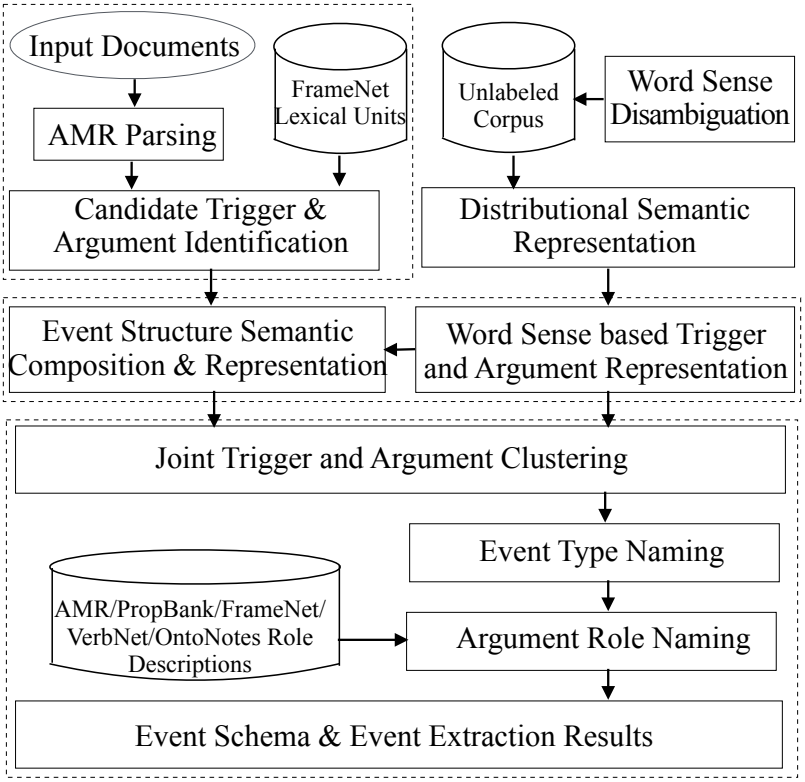

Figure 3.1: Liberal Event Extraction Overview.

Figure 3.1 illustrates the overall framework of Liberal Event Extraction. Given a set of input documents, we first extract semantic relations, apply WSD and learn word sense embeddings. Next, we identify candidate triggers and arguments.

For each event trigger, we apply a series of compositional functions to generate that trigger's event structure representation. Each function is specific to a semantic relation, and operates over vectors in the embedding space. Argument representations are generated as a by-product.

Trigger and argument representations are then passed to a joint constraint clustering framework. Finally, we name each cluster of triggers, and name each trigger's arguments using mappings between the meaning representation and semantic role descriptions in FrameNet, VerbNet [132] and Propbank [55].

We compare settings in which semantic relations connecting triggers to context words are

derived from three meaning representations: Abstract Meaning Representation (AMR) [133], Stanford Typed Dependencies [134], and FrameNet [53]. We derive semantic relations automatically for these three representations using CAMR [135], Stanford's dependency parser [136], and SEMAFOR [137], respectively.

3.3 BOTTOM-UP CANDIDATE EVENT TRIGGER AND ARGUMENT IDENTIFICATION

Although an event trigger may in principle be more than one word, more than 95% of the triggers consist of one single word. Furthermore, in the human annotated AMR data set, nearly 91.5% of triggers are parsed as verb concepts. Thus, trigger identification is simplified as the task of AMR parsing, incorporating the gazetteer created from FrameNet.

Given an input sentence, we first apply an AMR parser [135] to parse it. To maximize coverage, we consider all noun and verb concepts that can be linked to an OntoNotes [138] sense as candidate event triggers. In addition, if a word matches any lexical unit of a verb concept in FrameNet, we also consider it as a candidate event trigger. This can especially enrich nominal triggers such as "*war*", "*theft*" and "*pickpocket*".

For argument identification, we take E1 as an example. Figure 3.2 shows the events and argument annotations and AMR parsing results of E1. We can see that, most of the arguments are semantic related with triggers and can be identified based on the parsing results of AMR. On the other hand, AMR parsing results can help us discover much richer set of events and arguments. So, we carefully select 72 types of AMR relations[1] which are related to events, as shown in Table 3.2, and for each candidate event trigger, we collect all other concepts that are involved in these selected types of AMR relations as candidate arguments.

Categories	Relations
Core Roles	ARG0, ARG1, ARG2, ARG3, ARG4
Non-Core Roles	mod, location, poss, manner, topic, medium, instrument, duration, prep-X
Temporal	year, duration, decade, weekday, time
Spatial	destination, path, location

Table 3.2: Event-Related AMR Relations.

So in E1, "*killed*", "*injured*" and "*fighting*" are identified as candidate triggers, and three

[1]For relation details, see https://github.com/amrisi/amr-guidelines/blob/master/amr.md

concept sets are identified as candidate arguments: "{*Two Soldiers, very large missile*}", "{*one, Kut*}" and "{*Two Soldiers, Kut*}", as shown in Figure 3.2.

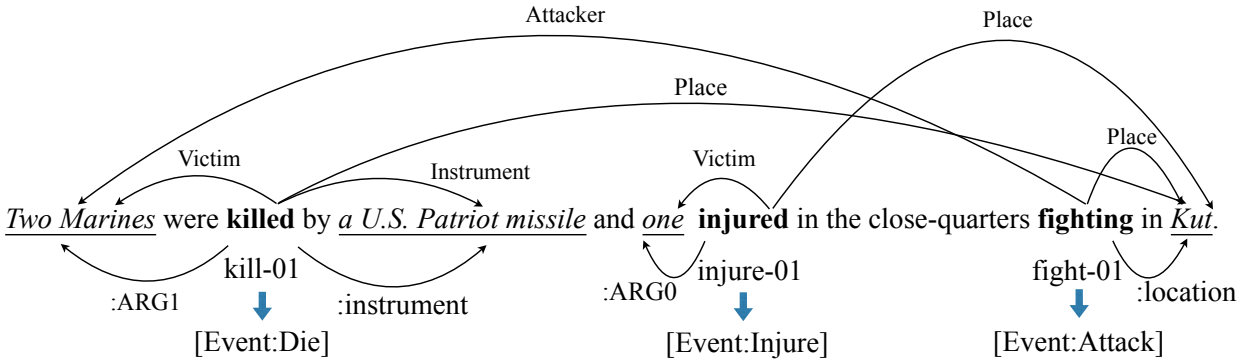

Figure 3.2: Event Trigger and Argument Annotations and AMR Parsing Results of *E1*.

3.4 TRIGGER SENSE AND ARGUMENT REPRESENTATION

Based on Hypothesis 3.1, each sense of a trigger may have a distinct type. Therefore we differentiate multiple senses and learn sense-based embeddings from a large data set, using the Continuous Skip-gram model [139]. Specifically, we first apply a state-of-the-art Word Sense Disambiguation (WSD) tool [140], which is trained on WordNet [141] and has achieved state-of-the art results on several SenseEval/SemEval English lexical-sample and all words tasks, to link each word to its sense in WordNet. Then based on a WordNet-OntoNotes sense mapping table [2] we map each trigger candidate to its OntoNotes sense and learn a distinct embedding for each sense. For arguments, we use their general lexical embedding as representations.

To capture the multi-word phrase embeddings, we compared two methods: (1) the model proposed by [142], which learned phrase embeddings directly by considering the phrase as a basic language unit, and (2) a simple element-based additive model ($z = x_1 + x_2 + ... + x_i$) [139], where z represents a phrase embedding and $x_1, x_2, ..., x_i$ represent the individual embeddings of the words in z. We found the latter performed better because Wikipedia is still too small to cover enough phrases (30% phrases in our test data appears 20 times or less in Wikipedia). Besides, method (2) can well capture multi-word expressions such as "*nuclear powered submarine*".

[2] https://catalog.ldc.upenn.edu/LDC2011T03

3.5 EVENT STRUCTURE COMPOSITION AND REPRESENTATION

Based on Hypothesis 3.2, the type of event does not only depend on the semantics of its trigger, but also depends on the arguments involved. Thus, we aim to exploit linguistic knowledge to represent event structures which incorporate event triggers, their arguments and inter-dependency relations. Many linguistic resources, including dependency parsing, semantic role labeling and VerbNet could be exploited. We will use AMR as an example to present our embedding composition method, and later we will compare the impact of various linguistic representations on event extraction.

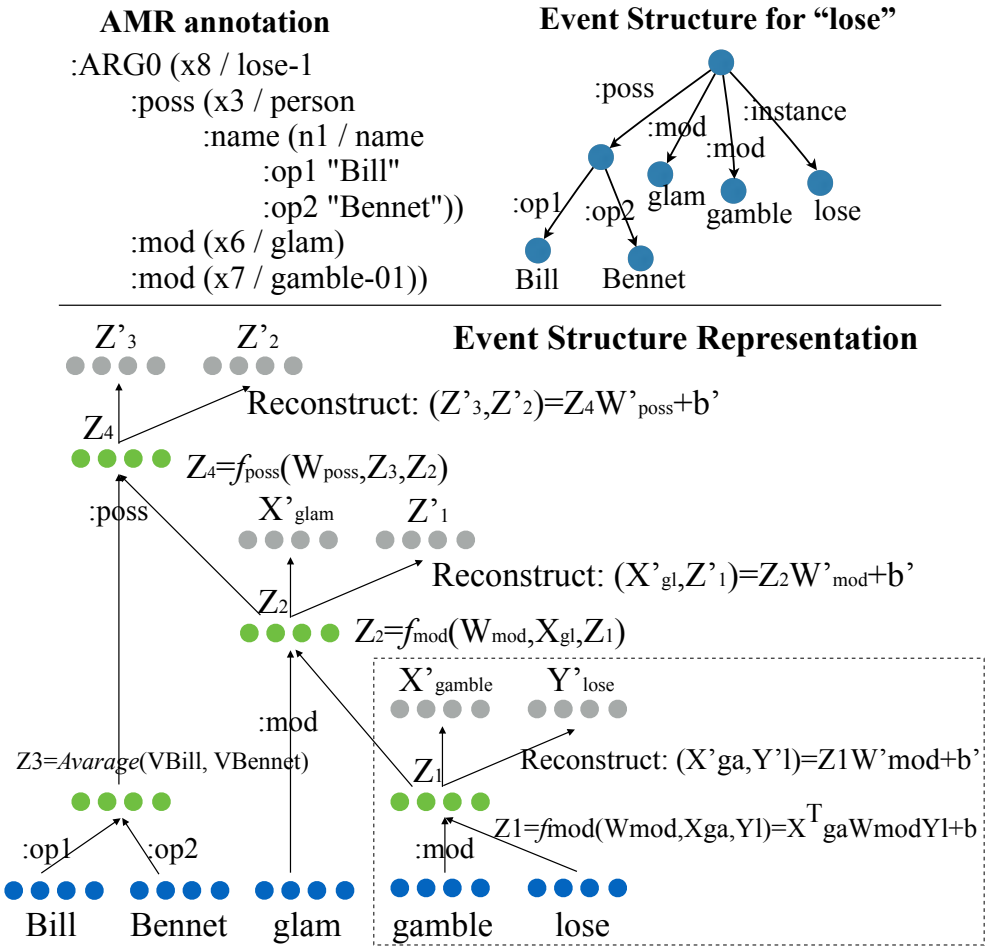

Figure 3.3: Partial AMR and Event Structure for *E2*.

Let's take *E2* as an example. Based on AMR and Table 3.2, we extract semantically related words for the event trigger with sense "*lose-1*" and construct the event structure for the whole event, as shown in Figure 3.3. In order to generate the representation for the whole event structure based on various semantic relations and arguments, we design a

Tensor based Recursive Auto-Encoder (TRAE) [143] framework to utilize a tensor based composition function for each AMR semantic relation and compose the event structure representation based on multiple functions.

Figure 3.3 shows an instance of a TRAE applied to the event structure. For each semantic relation type r, such as $:mod$, we define the output of a tensor product Z via the following vectorized notation:

$$Z = f_{mod}(X, Y, W_r^{[1:d]}, b) = [X; Y]^T W_r^{[1:d]} [X; Y] + b \tag{3.1}$$

where $W_{mod} \in \mathbb{R}^{2d \cdot 2d \cdot d}$ is a 3-order tensor, and $X, Y \in \mathbb{R}^d$ are two input word vectors. $b \in \mathbb{R}^d$ is the bias term. $[X; Y]$ denotes the concatenation of two vectors X and Y. Each slice of the tensor acts as a coefficient matrix for one entry Z_i in Z:

$$Z_i = f_{mod}(X, Y, W_r^{[i]}, b) = [X; Y]^T W_r^{[i]} [X; Y] + b_i \tag{3.2}$$

We use the average operation to compose the words connected by ":op" relations (e.g. "*Bill*" and "*Bennet*" in Figure 4).

After composing the vectors of X and Y, we apply an element-wise activation function sigmoid to the composed vector and generate the hidden layer representations Z. One way to optimize Z is to try to reconstruct the vectors X and Y by generating $X^{'}$ and $Y^{'}$ from Z, and minimizing the reconstruction errors between the input $V_I = [X, Y]$ and output layers $V_O = [X^{'}, Y^{'}]$. The error is computed based on Euclidean distance function:

$$E(V_I, V_O) = \frac{1}{2} ||V_I - V_O||^2 \tag{3.3}$$

For each pair of words X and Y, the reconstruction error back-propagates from its output layer to input layer through parameters $\Theta_r = (W_r^{'}, b_r^{'}, W_r, b_r)$. Let δ_O be the residual error of the output layer, and δ_H be the error of the hidden layer:

$$\delta_O = -(V_I - V_O) \cdot f_{\text{sigmoid}}^{'}(V_H^O) \tag{3.4}$$

$$\delta_H = (\sum_{k=1}^{d} \delta_O^k \cdot (W_r^{'k} + (W_r^{'k})^T) \cdot V_H^O) \cdot f_{\text{sigmoid}}^{'}(V_H^I) \tag{3.5}$$

where V_H^I and V_H^O denote the input and output of the hidden layer, and $V_H^O = Z$. $W_r^{'k}$ is the k^{th} slice of tensor $W_r^{'}$.

To minimize the reconstruction errors, we utilize gradient descent to iteratively update parameters Θ_r:

$$\frac{\partial E(\Theta_r)}{\partial W_r'^k} = \delta_O^k \cdot (V_H^O)^T \cdot V_H^O \tag{3.6}$$

$$\frac{\partial E(\Theta_r)}{\partial b_r'} = -(V_I - V_O) \cdot f'_{\text{sigmoid}}(V_H^O) \tag{3.7}$$

$$\frac{\partial E(\Theta_r)}{\partial W_r^k} = \delta_H^k \cdot (V_I)^T \cdot V_I \tag{3.8}$$

$$\frac{\partial E(\Theta_r)}{\partial b_r} = (\sum_{k=1}^{d} \delta_O^k \cdot (W_r'^k + (W_r'^k)^T) \cdot V_H^O) \cdot f'_{\text{sigmoid}}(V_H^I) \tag{3.9}$$

After computing the composition vector of Z_1 based on X and Y, for the next layer, it composes Z_1 and another new word vector such as X_{gl}. For each type of relation r, we randomly sample 2,000 pairs to train optimized parameters Θ_r. For each event structure tree, we iteratively repeat the same steps for each layer. The main advantage of this framework is that it can incorporate triggers, semantic relational arguments as well as various semantic relation types to generate multi-layer compositional event structure representations. For multiple arguments at each layer, we compose them in the order of their distance to the trigger: the closest argument is composed first.

3.6 EVENT TYPE SCHEMA INDUCTION

3.6.1 Joint Trigger and Argument Clustering

Based on the representation vectors generated above, we compute the similarity between each pair of triggers and arguments, and cluster them into types. We observe that, for two triggers t_1 and t_2, if their arguments have the same type and role, then they are more likely to belong to the same type. This is also true for two arguments when their semantic related *triggers* belong to the same type. For example, for event trigger "*capture*", when its ARG1 argument is a $Person$, such as "*Barzan Ibrahim Hasan al-Tikriti*" in E4, it usually refers to an *Arrest-Jail* event, just the same as triggers "*arrest*" and "*jail*", which also generally take $Person$ as ARG1 argument. However, when the ARG1 argument is a material object, such as "*Italian Ship*" in E5, it is much more likely to be a $Transfer\text{-}Ownership$ event. Therefore, we design a novel joint constraint co-clustering framework to encode constraints between two inter-dependent triggers and arguments so they can mutually enhance each other.

We first introduce a constraint function f, to enforce inter-dependent triggers and arguments to have coherent types:

$$f(\mathcal{P}_1, \mathcal{P}_2) = \log(1 + \frac{|\mathcal{L}_1 \cap \mathcal{L}_2|}{|\mathcal{L}_1 \cup \mathcal{L}_2|}) \tag{3.10}$$

where $\mathcal{P}_1$ and $\mathcal{P}_2$ are triggers. Elements of $\mathcal{L}_i$ are pairs of the form $(r, \mathrm{id}(a))$, where $\mathrm{id}(a)$ is the cluster ID for argument a that stands in relation r to $\mathcal{P}_i$. For example, let $\mathcal{P}_1$ and $\mathcal{P}_2$ be triggers "*capture*" and "*arrested*" (c.f. Figure 3.4). If *Barzan Ibrahim Hasan al-Tikriti* and *Ayman Sabawi Ibrahim* share the same cluster ID, the pair ($arg1$, id(Barzan Ibrahim Hasan al-Tikriti)) will be a member of $\mathcal{L}_1 \cap \mathcal{L}_2$. This argument overlap is evidence that "*capture*" and "*arrested*" have the same type. We define f where $\mathcal{P}_i$ are arguments, and elements $\mathcal{L}_i$ are defined analogously to above.

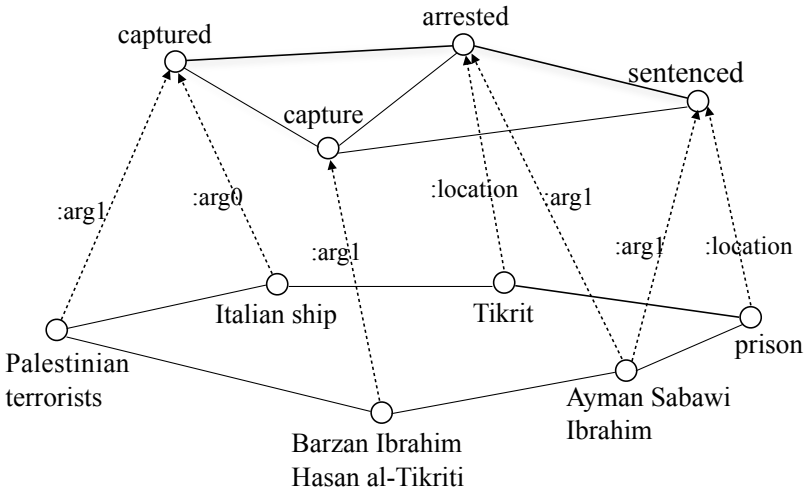

Figure 3.4: Joint Constraint Clustering for E3,4,5.

Given a trigger set T and their corresponding argument set A, we compute the similarity between two triggers t_1 and t_2 and two arguments a_1 and a_2 by:

$$sim(t_1, t_2) = \lambda \cdot sim_{\cos}(E_g^{t_1}, E_g^{t_2}) + (1 - \lambda) \cdot \frac{\Sigma_{r \in R_{t_1} \cap R_{t_2}} sim_{\cos}(E_r^{t_1}, E_r^{t_2})}{|R_{t_1} \cap R_{t_2}|} + f(t_1, t_2) \tag{3.11}$$

$$sim(a_1, a_2) = sim_{\cos}(E_g^{a_1}, E_g^{a_2}) + f(a_1, a_2) \tag{3.12}$$

where E_g^t represents the trigger sense vector and E_g^a is the argument vector. R_t is the AMR relation set in the event structure of t, and E_r^t denotes the vector resulting from the last application of the compositional function corresponding to the semantic relation r for

trigger t. λ is a regularization parameter that controls the trade-off between these two types of representations. In our experiment $\lambda = 0.6$.

We design a joint constraint clustering approach, which iteratively produces new clustering results based on the above constraints. To find a global optimum, which corresponds to an approximately optimal partition of the trigger set into K clusters $\mathcal{C}^T = \{\mathcal{C}_1^T, \mathcal{C}_2^T, ..., \mathcal{C}_K^T\}$, and a partition of the argument set into M clusters $\mathcal{C}^A = \{\mathcal{C}_1^A, \mathcal{C}_2^A, ..., \mathcal{C}_M^A\}$, we minimize the agreement across clusters and the disagreement within clusters:

$$\arg \min_{K_T, K_A, \lambda} O = (D_{\text{inter}}^T + D_{\text{intra}}^T) + (D_{\text{inter}}^A + D_{\text{intra}}^A) \tag{3.13}$$

$$D_{\text{inter}}^{\mathcal{P}} = \sum_{i \neq j = 1}^{K} \sum_{u \in \mathcal{C}_i^{\mathcal{P}}, v \in \mathcal{C}_j^{\mathcal{P}}} sim(\mathcal{P}_u, \mathcal{P}_v) \tag{3.14}$$

$$D_{\text{intra}}^{\mathcal{P}} = \sum_{i=1}^{K} \sum_{u,v \in \mathcal{C}_i^{\mathcal{P}}} (1 - sim(\mathcal{P}_u, \mathcal{P}_v)) \tag{3.15}$$

We incorporate the Spectral Clustering algorithm [144] into joint constraint clustering process to get the final optimized clustering results. The detailed algorithm is summarized in Algorithm 7.1.

3.6.2 Event Type and Argument Role Naming

After clustering, we assume each cluster represents a type and will assign a name for each type. For each trigger cluster, we utilize the trigger which is nearest to the centroid of the cluster as the event type name. In order to assign a role name to each argument, we map roles from AMR to available linguistic resources to name the arguments. For core roles (e.g., :ARG0, :ARG1) in AMR, we first link each concept from OntoNotes to FrameNet and VerbNet, and map AMR core roles to FrameNet roles and VerbNet roles[3]. Nearly 5% of AMR core roles can be mapped to FrameNet roles and 55% can be mapped to VerbNet roles. For the remaining concepts, we use the role descriptions from PropBank to name their roles. Table 3.3 shows some mapping examples. For non-core roles, we map them from AMR to FrameNet, as shown in Table 3.4.

[3]https://catalog.ldc.upenn.edu/LDC2013T19

Algorithm 3.1 Joint Constraint Clustering Algorithm

Input: Trigger set T, argument set A, their lexical embedding E_g^T, E_g^A, event structure representation E_R^T, and the minimal (K_T^{min}, K_A^{min}) and maximal (K_T^{max}, K_A^{max}) number of clusters for triggers and arguments;
Output: The optimal clustering results: $\mathcal{C}^T$ and $\mathcal{C}^A$;

- $O_{min} = \infty$, $\mathcal{C}^T = \varnothing$, $\mathcal{C}^A = \varnothing$
- For $K_T = K_T^{min}$ to $K_T = K_T^{max}$, $K_A = K_A^{min}$ to $K_A = K_A^{max}$
  - Clustering with Spectral Clustering Algorithm:
  - $\mathcal{C}_{curr}^T = spectral(T, E_g^T, E_R^T, K_T)$
  - $\mathcal{C}_{curr}^A = spectral(A, E_g^A, K_A)$
  - $O_{curr} = O(\mathcal{C}_{curr}^T, \mathcal{C}_{curr}^A)$
  - if $O_{curr} < O_{min}$
    - $O_{min} = O_{curr}$, $\mathcal{C}^T = \mathcal{C}_{curr}^T$, $\mathcal{C}^A = \mathcal{C}_{curr}^A$
  - while iterate time ≤ 10
    - $\mathcal{C}_{curr}^T = spectral(T, E_g^T, E_R^T, K_T, \mathcal{C}_{curr}^A)$
    - $\mathcal{C}_{curr}^A = spectral(A, E_g^A, K_A, \mathcal{C}_{curr}^T)$
    - $O_{curr} = O(\mathcal{C}_{curr}^T, \mathcal{C}_{curr}^A)$
    - if $O_{curr} < O_{min}$
      - O_{min}=O_{curr}, $\mathcal{C}^T = \mathcal{C}_{curr}^T$, $\mathcal{C}^A = \mathcal{C}_{curr}^A$
- return O_{min}, $\mathcal{C}^T$, $\mathcal{C}^A$;

3.7 EXPERIMENTS

3.7.1 Data

Since our approach is based on word embeddings, which need to be trained from a large corpus of unlabeled in-domain articles, we used the August 11, 2014 English Wikipedia dump to learn trigger sense and argument embeddings.

ACE (Automatic Content Extraction) [4] and ERE (Entities, Relations, Events) [145] provide comprehensive annotation standards for annotating Entities, Events and Relations. Table 3.5 shows the statistics of the types defined in ACE and ERE. Both programs released annotations for hundreds of articles from a wide variety of genres, including news and discussion forum. For event extraction evaluation, we choose a subset of ERE corpus (50 documents) which has perfect AMR annotations so we can compare the impact of perfect

[4] https://www.ldc.upenn.edu/collaborations/past-projects/ace

Concept	AMR Core Role	FrameNet Role	VerbNet Role	PropBank Description
fire.1	ARG0	Agent	Agent	Shooter
fire.1	ARG1	Projectile	Theme	Gun/projectile
fire.2	ARG0	Employer	Agent	Employer
fire.2	ARG1	Employee	Theme	Ex-employee
fire.2	ARG2	Task	Source	Job
extrude.1	ARG0		Agent	Extruder, agent
extrude.1	ARG1		Theme	Entity extruded
extrude.1	ARG2		Source	Extruded from
blood.1	ARG0			Agent
blood.1	ARG1			Theme, one bled

Table 3.3: Core Role Mapping Examples Between AMR and FrameNet, VerbNet, and PropBank.

AMR None-Core Role	FrameNet Role
topic	Topic
instrument	Instrument
manner	Manner
poss	Possessor
prep-for, prep-to, prep-on-behalf	Purpose
time, decade, year, weekday, duration	Time
mod, cause, prep-as	Explanation
prep-by, medium, path	Means
location, destination, prep-in	Place

Table 3.4: None-Core Role Mapping Between AMR and FrameNet.

AMR and system generated AMR. To compare with state-of-the-art event extraction on ACE2005 data, we follow the same evaluation setting in previous work [12, 57, 93] on data splitting and scoring metrics. We use 40 newswire documents as our test set. The detailed data statistics are presented in Table 3.6.

Evaluation Criteria and Metrics For event extraction task, we follow previous work [12, 57, 93] and use the following criteria to determine the correctness of an predicted event mention and evaluate the performances with mention-level Precision, Recall and F-measure:

- A trigger is correct if its event type and offsets match a reference trigger.

Programs	# of Entity Types	# of Relation Types	# of Relation Subtypes	# of Event Types	# of Event Subtypes
ACE	7	6	18	8	33
ERE	5	5	20	9	38

Table 3.5: Statistics of Entity, Relation, Event Types Covered by ACE and ERE

Data Sets	ACE	ERE
# of Documents	40	50
# of Sentences	799	1,053
# of Tokens	18,722	19,901

Table 3.6: Statistics of Data Sets

- An argument is correctly identified if its event type and offsets match any of the reference argument mentions.
- An argument is correctly identified and classified if its event type, offsets and role match any of the reference argument mentions.

3.7.2 Experimental Results for Schema Discovery

Event Type: Transport **S1**: The court official stated that on 18 March 2008 Luong stated to judges that she was hired by an unidentified *man* to **ship** the *heroin* to *Australia* in exchange for 15000 U.S. dollars. ***Event**: ship,* ***Arguments**: man(Agent), Austrialia(Destination),heroin(Theme)* **S2**: State media didn't identify the 2 convicts hanged in Zahedan but stated that *they* had been found guilty of **transporting** 5.25 kilograms of *heroin*. ***Event**: transporting, **Arguments**: they(Agent), heroin(Theme)*	**Event Type: Die** **S1**: Police in the strict communist country discovered his methamphetamine manufacturing plant disguised as a soap factory and sentenced *him* to **death** in *1997*. ***Event**: death, **Arguments**: him(Theme), 1997(Time)* **S2**: A newspaper report on January 1, 2008 that *Iran* **hanged** two convicted *drug traffickers* in the *south-eastern city of Zahedan*. ***Event**: hanged, **Arguments**: Iran(Agent), drug traffickers(Theme), southeastern city of Zahedan(Place)*
Event Type: Build **S1**: The **construction** of the *facility* started in *790000*, but stopped after the 910000 Soviet collapse when Tajikistan slid into a 5 year civil war that undermined its economy. ***Event**:construction,* ***Arguments**: facility(Product), 790000(Time)* **S2**: The closed *Soviet-era military facility* was **fou-nded** in *570000* and collects and analyzes all information gathered from Russia's military spy satellites. ***Event**: founded,* ***Arguments**: Soviet-era facility(Product), 570000(Time)*	**Event Type: Threaten** **S1**: *Colombian Government* was **alarmed** because uranium is the primary basis for generating weapons of mass destruction. ***Event**:alarmed, **Arguments**:Columbian Government(Experiencer)* **S2**: Cluster bomblets have been criticized by human rights groups because they kill indiscriminately and because unexploded *ordinance* poses a **threat** to *civilians* similar to that of land mines. ***Event**:threat,* ***Arguments**:ordinance(Cause), civilian(Experiencer)*

Figure 3.5: Example Output of the Event Schema.

Figure 3.5 shows some examples as part of the event schema discovered from the ERE data set. Each cluster denotes an event type, with a set of event mentions and sentences. Each event mention is also associated with some arguments and their roles. The annotations

for sample sentences may also serve as a corpus customized annotation guideline for event extraction.

Figure 3.6 shows an overview of the event schema (ontology) that our approach discovers. On event type level, by aggregating semantic coherent triggers into clusters, our approach extracts a set of abstracted event types, such as **Transport**, **Die**, **Build** and **Threaten**. In addition, our approach also discovers a set of argument roles for each event type, For example, for the Threaten event type, there are four argument roles, **Agent**, **Experiencer**, **Cause** and **Time**. It's worth noting that the argument roles are dependent on the types of argument mentions, which means, only few types' of entity mentions can play for a particular argument role, e.g., the **Product** of **Build** event can only be *Facility* rather than *Person* or **GPE**, which is consistent as the definition of human create event schemas.

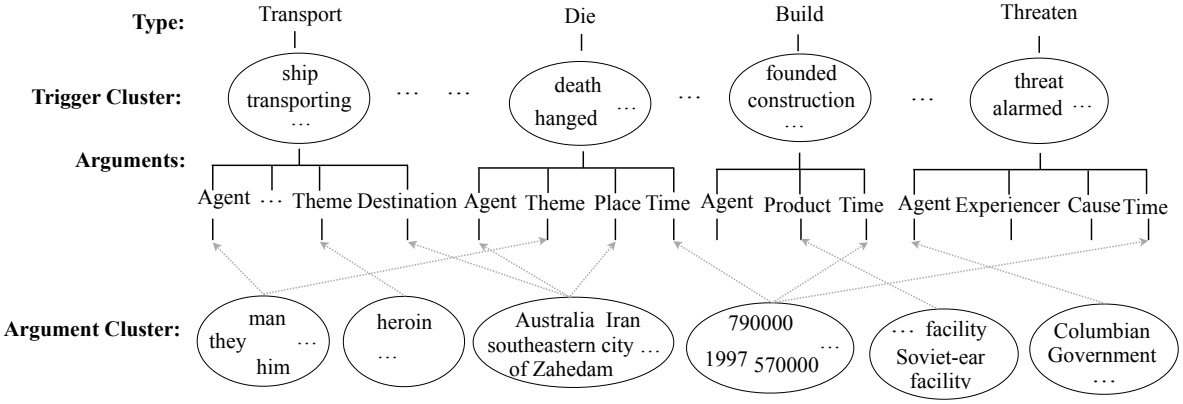

Figure 3.6: Overview of the Event Schema.

Data	ACE			ERE				
	Human	*System AMR*	*Overlap*	*Human*	*Perfect AMR*	*Overlap*	*System AMR*	*Overlap*
# of Events	440	2,395	331	580	3,765	517	2,498	477
# of Event Types	33	134	N/A	26	137	N/A	120	N/A
# of Arguments	883	4,361	587	1,231	6,195	919	4,288	801

Table 3.7: Schema Coverage Comparison on ACE and ERE.

Table 3.7 shows comparison on the coverage of event schema discovered by our approach with the predefined ACE and ERE event schemas. We can see that the coverage of both event types and argument roles by our approach is much higher than ACE and ERE. In order to evaluate the quality of the schema discovered by our approach, we asked two human

annotators to manually check whether each cluster can denote an event type, and whether the event type name is appropriate, and ask another annotator to do adjudication. About 84.3% (113/134) of all the clusters can denote event types, and the accuracy of event type naming is about 86.7% (98/113). There are two types of meaningless clusters which can not denote event types: (1) a cluster containing triggers of various types; (2) a cluster with triggers which cannot denote an event mention, e.g., include-01. Most of the type naming errors are due to that the centroid trigger that we selected cannot describe an event type, e.g., part-01 cannot well describe the event type for a cluster of triggers including part-01, join-01 group. Besides the types defined in ACE and ERE, our approach discovers many new event types such as **Build**, **Threaten** in Figure 3.5. Our approach can also discover new types of argument roles. For example, for Attack events, besides five types of existing arguments (Attacker, Target, Instrument, Time, and Place) defined in ACE, we also discover a new type of argument **Purpose**. For example, in "*The Dutch government, facing strong public anti-war pressure, said it would not commit fighting forces to the war against Iraq but added it supported the military* ***campaign*** *to* ***disarm Saddam****.*", "*disarm Saddam*" is identified as the **Purpose** for the **Attack** event triggered by "*campaign*".

3.7.3 Event Extraction for All Types

Since our approach can discover many new event types and arguments, besides the evaluation on ACE and ERE types, we also evaluate the overall performance of the whole event schema based on human annotations. To evaluate the performance of the whole event schema, we randomly sample 100 sentences from ERE data set and ask two linguistic experts to fully annotate the events and arguments. The inter-annotator agreement is over 83% for triggers and 79% for arguments. We manually map human annotated event types and argument roles with the event schema discovered by our approach and Table 3.8 shows the performance.

Method	Trigger Identification (%)			Trigger Typing (%)			Arg Identification (%)			Arg Typing (%)		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Perfect AMR	87.0	98.7	92.5	70.0	79.5	74.5	94.0	83.7	88.6	72.4	64.4	68.2
System AMR	93.0	67.2	78.0	69.8	50.5	58.6	95.7	59.6	73.4	68.9	42.9	52.9

Table 3.8: Overall Performance of Liberal Event Extraction on ERE data for All Event Types.

By error analysis we found that most of missing event triggers are multi-word expressions such as "*took office*" in "*The ruling Millennium Democratic Party (MDP), has suffered declining popularity since President Roh Moo-Hyun* ***took office*** *in February*" or words which are not verb or noun concepts, such as "*previously*" and "*formerly*" , which are **End-Position** events in "*As well as* ***previously*** *holding senior positions at Barclays Bank, BZW and Kleinwort Benson, McCarthy was* ***formerly*** *a top civil servant at the Department of Trade and Industry*". For argument identification, considering our approach heavily relies on semantic parsing results, it cannot identify some arguments that are implicitly related to triggers. For example, in "*Anti-corruption judge* ***Saul Pena*** *stated Montesinos has admitted to the abuse of authority* ***charge***", "*Saul Pena*" is not identified as a **Adjudicator** argument of event "*charge*" because it has no direct semantic relations with the event trigger. In addition, argument identification often requires some inference. For example, in *Anwar, 56, who this week completed four years in prison on a* ***corruption*** *charge, now faces an earliest possible* ***release*** *date of April 14, 2009 if he is given one third remission of his sentence for good behavior*", "*corruption*" is a **Crime** argument for the "*release*" event considering that it is a subsequent event of "*prison*", which holds the "*corruption*" argument.

3.7.4 Event Extraction for ACE/ERE Types

In order to compare with traditional event extraction, we conduct experiments on event types defined by ACE and ERE.

We manually assess whether an event discovered by our approach should be mapped to an ACE/ERE event or not for evaluation purpose. For arguments, we keep all core roles and Instrument/Possessor/Time/Place arguments. Using the mapped subset of events and arguments, we compare our approach with the following state-of-the-art supervised methods which are trained from 529 ACE documents or 336 ERE documents:

- DMCNN: A dynamic multi-pooling convolutional neural network based on distributed word representations [146].
- Joint: A structured perceptron model based on symbolic semantic features [13].
- LSTM: A long short-term memory neural network [147] based on distributed semantic features.

To evaluate the portability of each method on different datasets, we compare their performance on ERE and ACE. We train them on 529 ACE documents and 336 ERE documents for two experiments.

Tables 3.9 shows the results. On ACE events, both DMCNN and Joint methods outper-

Method	ERE: Trigger F_1 (%)			ERE: Arg F_1(%)			ACE: Trigger F_1 (%)			ACE: Arg F_1 (%)		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
LSTM	41.5	46.8	44.1	9.9	11.6	10.7	66.0	60	62.8	29.3	32.6	30.8
Joint	42.3	41.7	42.0	61.8	23.2	33.7	73.7	62.3	67.5	64.7	44.4	52.7
DMCNN	-	-	-	-	-	-	75.6	63.6	**69.1**	68.8	46.9	**53.5**
Liberal$_{PerfectAMR}$	79.8	50.5	**61.8**	48.9	32.9	**39.3**	-	-	-	-	-	-
Liberal$_{SystemAMR}$	88.5	42.6	57.5	47.6	30.0	36.8	80.7	50.1	61.8	51.9	39.4	44.8

Table 3.9: Performance on ERE and ACE events.

form our approach for trigger and argument extraction. However, when moving to ERE event schema, although re-trained based on ERE labeled data, their performance still degrades significantly. These previous methods heavily rely on the quality and quantity of the training data. So when the training data is not adequate (the ERE training documents contain 1,068 events and 2,448 arguments, while ACE training documents contain more than 4,700 events and 9,700 arguments), the performance is low. In contrast, our approach doesn't rely on any training corpus and it can automatically identify events, arguments and assign types/roles, so when the definition of event schema is changed, the performance will not be affected.

3.7.5 Event Extraction for Biomedical Domain

To demonstrate the portability of our approach to a new domain, we take the biomedical domain as a case study. We conduct our experiment on 14 biomedical articles (755 sentences) with perfect AMR annotations [148]. We utilize a word2vec model[5] trained from all paper abstracts from PubMed[6] and full-text documents from the PubMed Central Open Access subset. To evaluate the performance, we randomly sample 100 sentences and ask a biomedical scientist to assess the correctness of each event and argument role. Our approach achieves 83.1% precision on trigger labeling (619 events in total) and 78.4% precision on argument labeling (1,124 arguments in total). It demonstrates that our approach can be rapidly adapted to a new domain and discover domain-rich event schema. An example schema for an event type "*Dissociate*" is shown in Figure 3.7.

From the results, we have several observations that : (1) All event types in biomedical we discovered cannot be mapped to ACE/ERE event schema. (2). Most of events in biomedical

[5]http://bio.nlplab.org/

[6]http://www.ncbi.nlm.nih.gov/pubmed

Event Type: Dissociate

S1: Ras acts as a molecular switch that is activated upon GTP loading and deactivated upon **hydrolysis** of GTP to GDP.

***Event**: hydrolysis **Arguments**:GTP (Patient), (GDP) (Result)*

S2: Activation requires **dissociation** of protein-bound GDP , an intrinsically slow process that is accelerated by guanine nucleotide exchange factors.

***Event**: dissociation **Arguments**: GDP (Patient)*

S3: His - ubiquitinated proteins were purified by Co2+ metal affinity chromatography in 8M urea **denaturing** conditions.

***Event**: denaturing **Arguments**: proteins(Patient)*

Figure 3.7: Example Output of the Discovered Biomedical Event Schema.

are unique and unambiguous, and the events with the same string often refer to the same type of event. This observation is also demonstrated by the experiment results: without WSD, the precision of events typing is still over 80%. We utilize the ambiguity measure defined in [149] as the criteria to demonstrate the ambiguity degree of general domain (16.2%) and biomedical domain (6.7%).

$$ambiguity = \frac{\#event\ Strings\ belong\ to\ more\ than\ one\ cluster}{\#event\ Strings} \tag{3.16}$$

3.8 DISCUSSION: IMPACT OF SEMANTIC INFORMATION AND MEANING REPRESENTATIONS

In this work, we utilize AMR (Abstract Meaning Representation) to generate the compositional representations. Many linguistic resources can be used to capture symbolic semantics. Compared with dependency parsing and semantic role labeling, AMR has three advantages: (1) both syntactic and semantic information can be captured based on rich types of semantic relations from specific contexts, which is crucial for many IE tasks, while dependency parsing mainly focused on syntactic information; (2) AMR relations are much more fine-grained, which could be helpful for argument role induction; (3) Available AMR parsers are trained based on dependency parsing results.

To evaluate the impact of the symbolic and semantic representations, we also design several baselines and evaluate on the 100 ERE sentences with ground truth.

Table 3.10 shows the efffectiveness of each type of representation. We can see that: Combining the WSD based lexical representation and event structure representation together, our approach can get much better performance than based on single type of representation, which demonstrate the effectiveness of our Hypothesis 3.1 and Hypothesis 3.2. The

Method	Trigger F_1 (%)			Arg F_1 (%)		
	P	R	F_1	P	R	F_1
Liberal	70	79.5	74.5	72.4	64.4	68.2
w/o Structure Representation	52.8	59.4	55.9	52.1	48.0	50.0
w/o WSD	62.8	57.4	60.1	61.9	50.3	55.5
w/o None-Core Roles	61.5	72.2	66.5	61.3	58.0	59.6
w/o Core Roles	57.3	49.7	53.2	63.6	49.5	55.7
Replace AMR with Dependency Parsing	50.1	39.8	44.3	28.1	10.1	14.8

Table 3.10: Impact of Representations in Liberal Event Extraction on ERE data.

performance of our approach based on core AMR roles is much better than that based on none-core roles, which demonstrates that the core AMR roles are more meaningful for event extraction. What's more, when using dependency parsing to replace AMR, we manually map dependency relations to FrameNet roles to determine the argument roles and find the performance decreases significantly. Compared with dependency relations, the fine-grained AMR semantic relations such as *:location*, *:manner*, *:topic*, *:instrument* are much more informative to infer the argument roles. For example, in sentence "*Approximately 25 kilometers southwest of Sringar 2 militants were killed in a second gun battle.*", "*gun*" is identified as an **Instrument** for "*battle*" event based on the AMR relation :instrument. In contrast, dependency parsing identifies "*gun*" as a *compound* modifier of "*battle*".

3.9 SUMMARY

Traditional information extraction approaches heavily rely on a set of predefined types and human annotated data, thus suffer from high cost and low portability. In this chapter, we discuss a novel Liberal event extraction framework which combines the merits of symbolic semantics, e.g., Abstract Meaning Representation, and distributed semantics, including word sense representations and local context representations learned from a tree-based recursive auto-encoder. Experiments on news and biomedical domain demonstrate that this framework can discover explicitly defined rich event schemas which cover not only most types in existing manually defined schemas, such as ACE and ERE, but als o new event types (e.g., Threaten, Building) and argument roles (e.g., Purpose of Attack event type). The granularity of event types is also customized for specific input corpus, and it can produce high-quality event annotations simultaneously without using annotated training data.

The general philosophy of liberal IE framework is to leverage rich representations, including word senses, local contexts, or even external knowledge, to better disambiguate the meaning of the knowledge elements and further induce high-level representations, e.g., type schema. This framework has been successfully applied to entity and event extraction tasks, and can also be adapted to other schema guided tasks.

However, we also notice several remaining limitations of this liberal IE framework: (1) it also detects a lot of nonsensical event triggers, such as know, feel, say, hear, etc.; (2) it misses a lot of multi-token event mentions (e.g., take place) and adverb event mentions (e.g., previously and formerly); (3) in some cases, the semantics of event mentions is not enough to indicate their types, for example, hire and resign share very similar semantic representations, however they indicate different state changes so they still should belong to different types.

CHAPTER 4: ZERO SHOT INFORMATION EXTRACTION: EXTENDING EVENT TYPES FROM 30+ TO 1300+

The Liberal Information Extraction framework can automatically extract facts as well as a type schema for a given corpora. However, in many practical applications, customers usually require to extract certain types of event mentions for a specific domain, while there is no or very limited in-domain annotations. In this case, liberal IE framework is not very suitable since it's not easy to automatically map the automatically induced types to a predefined target ontology. Thus, beyond absolute unsupervised information extraction, we design a zero shot transfer learning framework, which can take advantage of existing annotated data for any types, and transfer the knowledge from the old types to any new types.

4.1 MOTIVATIONS

Traditional supervised methods have typically modeled event extraction as a classification problem, by assigning event triggers to event types from a pre-defined fixed set. These methods rely heavily on manual annotations and features specific to each event type, and thus are not easily adapted to new event types without extra annotation effort. Handling new event types may entail starting over, without being able to re-use annotations for previous event types.

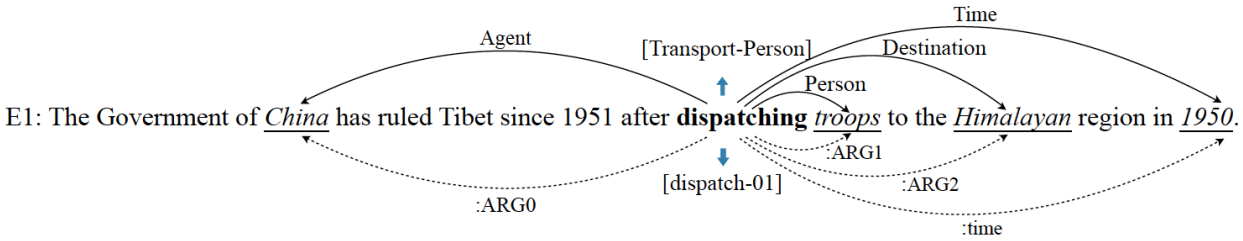

Figure 4.1: Event Mention Example: **dispatching** is the trigger of a *Transport-Person* event with four arguments.

To make event extraction more feasible for practical emergent settings, we take a fresh look at this task. We observe that each event mention has a structure consisting of a candidate trigger and arguments, with corresponding pre-defined name labels for the event type and argument roles. Let's consider two example sentences:

E1. The Government of *China* has ruled Tibet since 1951 after **dispatching** *troops* to the *Himalayan* region in *1950*.

E2. Iranian state television stated that the **conflict** between the *Iranian police* and the drug *smugglers* took place near the town of *mirjaveh*.

In E1, as also diagrammed in Figure 4.1, **dispatching** is the trigger for the event mention of type *Transport_Person* and in E2, **conflict** is the trigger for the event mention of type *Attack*. We make use of Abstract Meaning Representations (AMR) [133] to identify the candidate arguments and construct event mention structures as shown in Figure 4.2. Figure 4.2 also shows event type structures defined in the Automatic Content Extraction (ACE) guideline [1]. We can see that, a trigger and its event type usually share similar lexical semantic meaning. Besides the lexical semantics that relates a trigger to its type, their structures also tend to be similar : a *Transport_Person* event typically involves a *Person* as its *patient* role, while an *Attack* event involves a *Person* or *Location* as an *Attacker*. This observation matches the theory by [58]: "the semantics of an event structure can be generalized and mapped to event mention structures in a systematic and predictable way".

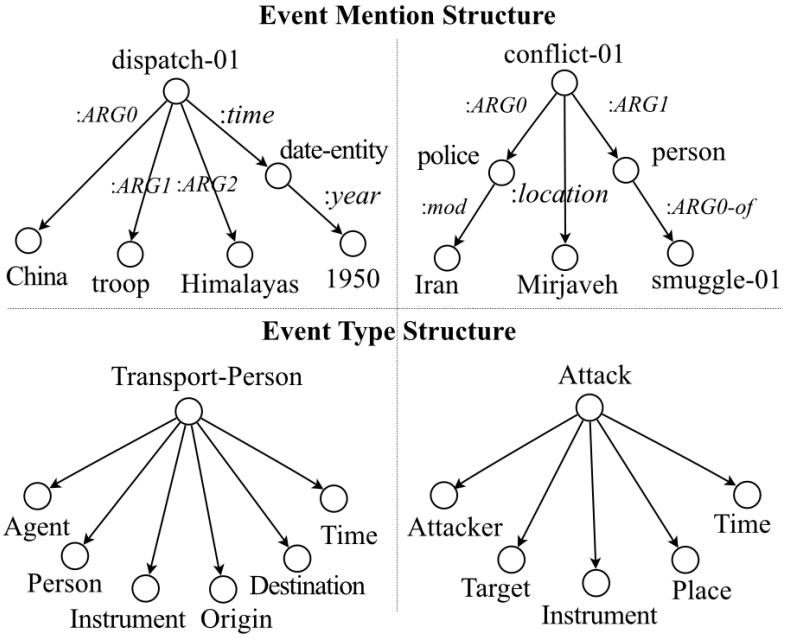

Figure 4.2: Examples of Event Mention Structures and Type Structures from ACE.

Inspired by this theory, for the first time, we model event extraction as a grounding problem, by mapping each mention to its semantically closest event type. Given an event ontology, where each event type structure is well defined (e.g., argument roles), we call the event types with annotated event mentions as *seen* types, while those without annotations as *unseen* types. Our goal is to learn a generic mapping function independent of event

[1] https://en.wikipedia.org/wiki/Automatic_content_extraction

types, which can be trained from annotations for a limited number of seen event types and further used for any new unseen event types. We design a transferable neural architecture, which jointly learns and maps the structural representations of event mentions and types into a shared semantic space, by minimizing the distance between each event mention and its corresponding type. For event mentions with *unseen* types, their structures will be projected into the same semantic space using the same framework and assigned types with top-ranked similarity values.

To summarize, to apply our new zero-shot transfer learning framework to any new unseen event types, we only need (1) a structured definition of the unseen event type (its type name along with role names for its arguments); and (2) some annotations for one or a few seen event types. Without using any manual annotations for the new unseen types, our framework achieves performance comparable to supervised methods trained from a substantial amount of training data for the same types.

4.2 APPROACH OVERVIEW

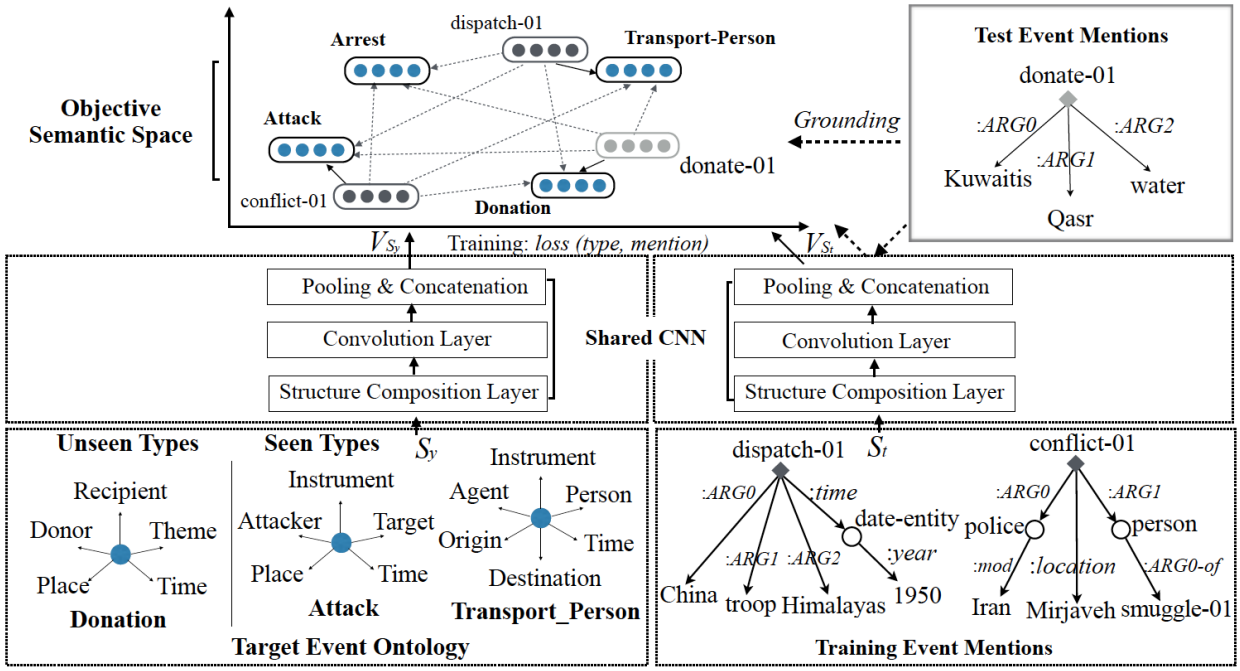

Figure 4.3: Architecture Overview. The blue circles denote event types and event type representations. The dark grey diamonds and circles denote triggers and trigger representations from training set. The light grey diamonds and circles denote triggers and trigger representations from testing set.

Given a sentence s, we start by identifying candidate triggers and arguments based on AMR parsing [150]. For the example shown in Figure 4.1, we identify *dispatching* as a trigger, and its candidate arguments: *China*, *troops*, *Himalayan* and *1950*. Then we use our new neural architecture depicted in Figure 4.3 to classify triggers into event types. The argument classification follows the same pipeline.

For each trigger t, e.g., *dispatch-01*, we determine its type by comparing its semantic representation with that of any event type in the event ontology. In order to incorporate the contexts into the semantic representation of t, we build a structure S_t using AMR as shown in Figure 4.3. Each structure is composed of a set of tuples, e.g, $\langle$*dispatch-01*, *:ARG0*, *China*$\rangle$. We use a matrix to represent each AMR relation, composing its semantics with two concepts for each tuple. As CNN can capture sequence level feature representation, we and feed all tuple representations into a CNN to generate a dense vector representation V_{S_t} for the event mention structure.

Given a target event ontology, for each type y, e.g., *Transport_Person*, we construct a type structure S_y consisting of its predefined roles, and use a tensor to denote the implicit relation between any type and argument role. We compose the semantics of type and argument role with the tensor for each tuple, e.g., $\langle$*Transport_Person*, *Destination*$\rangle$. Then we generate the event type structure representation V_{S_y} using the same CNN. By minimizing the semantic distance between *dispatch-01* and *Transport_Person* using V_{S_t} and V_{S_y}, we jointly map the representations of event mention and event types into a shared semantic space, where each mention is closest to its annotated type.

After training, the compositional functions and CNNs can be further used to project any new event mention (e.g., *donate-01*) into the semantic space and find its closest event type (e.g., *Donation*). In the next sections we will elaborate each step in great detail.

4.3 TRIGGER AND TYPE STRUCTURE COMPOSITION

As Figure 4.3 shows, for each candidate trigger t, we construct its event mention structure S_t based on its candidate arguments and AMR parsing. For each type y in the target event ontology, we construct a structure S_y by including its pre-defined roles and using its type as the root.

Each S_t or S_y is composed of a collection of tuples. For each event mention structure, a tuple consists of two AMR concepts and an AMR relation. For each event type structure, a tuple consists of a type name and an argument role name. Next we will describe how to compose semantic representations for event mention and event type respectively based on these structures.

Event Mention Structure For each tuple $u = \langle w_1, \lambda, w_2 \rangle$ in an event mention structure, we use a matrix to represent each AMR relation λ, and compose the semantics of λ between two concepts w_1 and w_2 as:

$$V_u = [V_{w_1}^{'}; V_{w_2}^{'}] = f([V_{w_1}; V_{w_2}] \cdot M_\lambda) \tag{4.1}$$

where V_{w_1}, $V_{w_2} \in \mathbb{R}^d$ are the vector representations of words w_1 and w_2. d is the dimension size of each word vector. [;] denotes the concatenation of two vectors. $M_\lambda \in \mathbb{R}^{2d \times 2d}$ is the matrix representation for AMR relation λ. V_u is the composition representation of tuple u, which consists of two updated vector representations $V_{w_1}^{'}$, $V_{w_2}^{'}$ for w_1 and w_2 by incorporating the semantics of λ.

Event Type Structure For each tuple $u^{'} = \langle y, r \rangle$ in an event type structure, where y denotes the event type and r denotes an argument role, following Socher et al. [151], we assume an implicit relation exists between any pair of type and argument, and use a single and powerful tensor to represent the implicit relation:

$$V_{u'} = [V_y^{'}; V_r^{'}] = f([V_y; V_r]^T \cdot U^{[1:2d]} \cdot [V_y; V_r]) \tag{4.2}$$

where V_y and V_r are vector representations for y and r. $U^{[1:2d]} \in \mathbb{R}^{2d \times 2d \times 2d}$ is a 3-order tensor. $V_u^{'}$ is the composition representation of tuple $u^{'}$, which consists of two updated vector representations $V_y^{'}$, $V_r^{'}$ for y and r by incorporating the semantics of their implicit relation $U^{[1:2d]}$.

4.4 TRIGGER AND ARGUMENT CLASSIFICATION

4.4.1 Trigger Classification for Seen Types

Both event mention and event type structures are relatively simple and can be represented with a set of tuples. CNNs have been demonstrated effective at capturing sentence level information by aggregating compositional n-gram representations. In order to generate structure-level representations, we use CNN to learn to aggregate all edge and tuple representations.

Input layer is a sequence of tuples, where the order of tuples is from top to bottom in the structure. Each tuple is represented by a $d \times 2$ dimensional vector, thus each mention

structure and each type structure are represented as a feature map of dimensionality $d \times 2h^*$ and $d \times 2p^*$ respectively, where h^* and p^* are the maximal number of tuples for event mention and type structures. We use zero-padding to the right to make the volume of all input structures consistent.

Convolution layer Take S_t with h^* tuples: $u_1, u_2, ..., u_{h^*}$ as an example. The input matrix of S_t is a feature map of dimensionality $d \times 2h^*$. We make c_i as the concatenated embeddings of n continuous columns from the feature map, where n is the filter width and $0 < i < 2h^* + n$. A convolution operation involves a filter $W \in \mathbb{R}^{nd}$, which is applied to each sliding window c_i:

$$c_i^{'} = \tanh(W \cdot c_i + b) \tag{4.3}$$

where $c_i^{'}$ is the new feature representation, and $b \in \mathbb{R}^d$ is a biased vector. We set filter width as 2 and stride as 2 to make the convolution function operate on each tuple with two input columns.

Max-Pooling: All tuple representations $c_i^{'}$ are used to generate the representation of the input sequence by max-pooling.

Learning: For each event mention t, we name the correct type as *positive* and all the other types in the target event ontology as *negative*. To train the composition functions and CNN, we first consider the following hinge ranking loss:

$$L_1(t, y) = \sum_{j \in Y,\ j \neq y} \max\{0, m - C_{t,y} + C_{t,j}\} \tag{4.4}$$

$$C_{t,y} = \cos([V_t; V_{S_t}], [V_y; V_{S_y}]) \tag{4.5}$$

where y is the positive event type for t. Y is the type set of the target event ontology. $[V_t; V_{S_t}]$ denotes the concatenation of representations of t and S_t. j is a negative event type for t from Y. m is a margin. $C_{t,y}$ denotes the cosine similarity between t and y.

The hinge loss is commonly used in zero-shot visual object classification task. However, it tends to overfit the seen types in our experiments. Compared with zero-shot visual object classification, our task has much fewer seen types for training, for example, ACE defined 33 event types, whereas there are usually more than 1,000 seen types in visual object classification, thus the model will be easier to be biased to the limited seen types. While clever data augmentation can help alleviate overfitting, we propose to add "negative" event mentions into the training process. Here a "negative" event mention means that the mention has no

positive event type among all seen types, namely it belongs to *Other*. Though we don't know the exact type of the negative event mentions, we know that their types must be from unseen types rather than seen ones. To encourage the negative event mentions to be mapped to unseen types, we design a new loss function as follows:

$$L_1^d(t,y) = \begin{cases} \max\limits_{j \in Y, j \neq y} \max\{0, m - C_{t,y} + C_{t,j}\}, & y \neq Other \\ \max\limits_{j \in Y', j \neq y'} \max\{0, m - C_{t,y'} + C_{t,j}\}, & y = Other \end{cases} \tag{4.6}$$

where Y is the type set of the event ontology. $Y^{'}$ is the seen type set. y is the annotated type. $y^{'}$ is the type which ranks the highest among all event types for event mention t, while t belongs to *Other*.

By minimizing L_1^d, we can learn the optimized model which can compose structure representations and map both event mention and types into a shared semantic space, where the positive type ranks the highest for each mention.

4.4.2 Argument Classification for Seen Types

For each mention, we map each candidate argument to one of the pre-defined roles following the same pipeline. Each argument candidate is matched to a specific role based on the semantic similarity of the argument path. Take E1 as an example. *China* is matched to *Agent* based on the semantic similarity between *dispatch-01*→ *:ARG0*→ *China* and *Transport-Person*→*Agent*.

Given a trigger t and a candidate argument a, we first extract a path $S_a = (u_1, u_2, ..., u_p)$, which connects t and a and consists of p tuples. Each predefined role r is also represented as a structure by incorporating the event type, $S_r = \langle y, r \rangle$. We apply the same framework to take the sequence of tuples contained in S_a and S_r into a weight-sharing CNN to rank all possible roles for a.

$$L_2^d(a,r) = \begin{cases} \max\limits_{j \in R_y, j \neq r} \max\{0, m - C_{a,r} + C_{a,j}\} & r \neq Other \\ \max\limits_{j \in R_{Y'}, j \neq r'} \max\{0, m - C_{a,r'} + C_{a,j}\} & r|y = Other \end{cases} \tag{4.7}$$

where R_y and $R_{Y'}$ are the set of argument roles which are predefined for trigger type y and all seen types $Y^{'}$. r is the annotated role and $r^{'}$ is the argument role which ranks the highest for a when a or y is annotated as *Other*.

In our experiments, the trigger and argument annotations are not balanced. Therefore, we sample various size of "negative" training data for trigger and argument labeling respectively. In the following section, we describe how the negative training instances are generated.

4.4.3 Zero-Shot Classification for Unseen Types

During test, given a new event mention $t^{'}$, we compute its mention structure representation for $S_{t'}$ and all event type structure representations for $S_Y = \{S_{y_1}, S_{y_2}, ..., S_{y_n}\}$ using the same parameters trained from seen types. Then we rank all event types based on their similarity scores with mention $t^{'}$. The top ranked prediction for $t^{'}$ from the event type set, denoted as $\widehat{y}(t^{'}, 1)$, is given by:

$$\widehat{y}(t^{'}, 1) = \arg\max_{y \in Y} \cos([V_{t'}; V_{S_{t'}}], [V_y; V_{S_y}]) \tag{4.8}$$

Moreover, $\widehat{y}(t^{'}, k)$ denotes the k^{th} most probable event type predicted for $t^{'}$. We will investigate the event extraction performance based on the top-k predicted event types.

After determining the type $y^{'}$ for mention $t^{'}$, for each candidate argument, we adopt the same ranking function to find the most appropriate role from the role set defined for $y^{'}$.

4.5 EXPERIMENTS

4.5.1 Hyperparameters

We use the English Wikipedia dump to learn trigger sense and argument embeddings based on the Continuous Skip-gram model [139]. Table 7.3 shows the hyper-parameters we use to train models.

Parameter Name	Value
Word Sense Embedding Size	200
Initial Learning Rate	0.1
# of Filters in Convolution Layer	500
Maximal # of Tuples for Mention Structure	10
Maximal # of Tuples for Argument Path	5
Maximal # of Tuples for Event Type Structure	5
Maximal # of Tuples for Argument Role Path	1

Table 4.1: Hyper-Parameters.

4.5.2 ACE Event Classification

We first use ACE event schema [2] as our target event ontology and assume the boundaries of triggers and arguments are given. Of the 33 ACE event types, we select the top-N most popular event types from ACE05 data as "seen" types, and use 90% event annotations of these for training and 10% for development. N is set as 1, 3, 5, 10 respectively. We test the zero-shot classification performance on the annotations for the remaining 23 unseen types. Table 4.2 shows the types that we selected for training in each experiment setting.

Setting	N	Seen Types for Training/Dev
A	1	Attack
B	3	Attack, Transport, Die
C	5	Attack, Transport, Die, Meet, Arrest-Jail
D	10	Attack, Transport, Die, Meet, Sentence, Arrest-Jail, Transfer-Money, Elect, Transfer-Ownership, End-Position

Table 4.2: Seen Types in Each Experiment Setting.

Setting Index	Training			Development		Test		
	# of Types, Roles	# of Events	# of Arguments	# of Events	# of Arguments	# of Types/Roles	# of Events	# of Arguments
A	1, 5	953/900	894/1,097	105/105	86/130	23/59	753	879
B	3, 14	1,803/1,500	2,035/1,791	200/200	191/237			
C	5, 18	2,033/1,300	2,281/1,503	225/225	233/241			
D	10, 37	2537/700	2,816/879	281/281	322/365			

Table 4.3: Statistics for Positive/Negative Instances in Training, Dev, and Test Sets for Each Experiment.

The negative event mentions and arguments that belong to *Other* are sampled from the output of the system developed by Huang et al. [152] based on ACE05 training sentences, which groups all candidate triggers and arguments into clusters based on semantic representations and assigns a type/role name to each cluster. We sample the negative event mentions from the clusters (e.g., *Build*, *Threaten*) which cannot be mapped to ACE event types. We sample the negative arguments from the arguments of negative event mentions. Table 4.3 shows the statistics of the training, development and testing data sets.

[2] ACE event schema specification is at: https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/english-events-guidelines-v5.4.3.pdf

Setting	Method	Hit@k Trigger Classification (%)			Hit@k Argument Classification (%)		
		k=1	k=3	k=5	k=1	k=3	k=5
	WSD-Embedding	1.7	13.0	22.8	2.4	2.8	2.8
A	Our Approach	4.0	23.8	32.5	1.3	3.4	3.6
B		7.0	12.5	36.8	3.5	6.0	6.3
C		20.1	34.7	46.5	9.6	14.7	15.7
D		33.5	51.4	68.3	14.7	26.5	27.7

Table 4.4: Comparison between Structural Representation (Our Approach) and Word Sense Embedding based Approaches on Hit@K Accuracy (%) for Trigger and Argument Classification.

To show the effectiveness of structural similarity in our approach, we design a baseline, WSD-Embedding, which directly maps event mentions and arguments to their candidate types and roles using our pre-trained word sense embeddings. Table 4.4 shows the comparison. We can see that the structural similarity is much more effective than lexical similarity for both trigger and argument classification. Also, as the number of seen types in training increases, the performance of the transfer model is improved.

We further evaluate the performance of our transfer approach on similar and distinct unseen types. The 33 subtypes defined in ACE fall within 8 coarse-grained main types, such as *Life* and *Justice*. Each subtype belongs to one main type. Subtypes that belong to the same main type tend to have similar structures. For example, *Trial-Hearing* and *Charge-Indict* have the same set of argument roles. For training our transfer model, we select 4 subtypes of *Justice*: **Arrest-Jail**, **Convict**, **Charge-Indict**, **Execute**. For testing, we select 3 other subtypes of *Justice*: *Sentence, Appeal, Release-Parole*. Additionally, we also select one subtype from each of the other seven main types for comparison. Table 4.5 shows that, when testing on a new unseen type, the more similar it is to the seen types, the better performance is achieved.

4.5.3 ACE Event Identification and Classification

ACE2005 corpus includes the richest event annotations for 33 types. However, in real scenario, there may be thousands of event types of interest. In order to enrich the target event ontology and assess our transferable neural architecture on a large number of unseen types when trained on limited annotations of seen types, we manually construct a new event ontology which combines 33 ACE event types and argument roles, and 1,161 frames from

Type	Subtype	Hit@k Trigger Classification		
		1	3	5
Justice	Sentence	68.3	68.3	69.5
Justice	Appeal	67.5	97.5	97.5
Justice	Release-Parole	73.9	73.9	73.9
Conflict	Attack	26.5	44.5	46.7
Transaction	Transfer-Money	48.4	68.9	79.5
Business	Start-Org	0	33.3	66.7
Movement	Transport	2.6	3.7	7.8
Personnel	End-Position	9.1	50.4	53.7
Contact	Phone-Write	60.8	88.2	90.2
Life	Injure	87.6	91.0	91.0

Table 4.5: Performance on Various Types Using Justice Subtypes for Training

Method	Trigger Identification			Trigger Identification + Classification			Arg Identification			Arg Identification + Classification		
	P	R	F	P	R	F	P	R	F	P	R	F
Supervised LSTM	94.7	41.8	58.0	89.4	39.5	54.8	47.8	22.6	30.6	28.9	13.7	18.6
Supervised Joint	55.8	67.4	61.1	50.6	61.2	55.4	36.4	28.1	31.7	33.3	25.7	29.0
Transfer	85.7	41.2	55.6	75.5	36.3	49.1	28.2	27.3	27.8	16.1	15.6	15.8

Table 4.6: Event Trigger and Argument Extraction Performance (%) on Unseen ACE Types.

FrameNet except for the most generic frames such as *Entity* and *Locale*. Some ACE event types can be easily aligned to frames, e.g., *Die* is aligned with *Death*. Some frames are instead more accurately treated as inheritors of ACE types, such as *Suicide-Attack*, which inherits from *Attack*. We manually mapped the selected frames to ACE types.

We compare our approach with the following state-of-the-art *supervised* methods:

- LSTM: A long short-term memory neural network [147] based on distributed semantic features, similar to [153].
- Joint: A structured perceptron model based on symbolic semantic features [13].

For our approach, we follow the experiment setting D in the previous section, using the same training and development data sets for the 10 seen types, but target at all of the 1,194

event types in our new event ontology instead of just the 33 ACE event types. For evaluation, we sample 150 sentences from the remaining ACE05 data, including 129 annotated event mentions for the 23 unseen types. For both LSTM and Joint approaches, we use the entire ACE05 annotated data for 33 ACE event types for training except for the held-out 150 evaluation sentences.

We first identify the candidate triggers and arguments, then map each of them to the target event ontology. We evaluate our model on extracting the event mentions which are classified into 23 testing ACE types. Table 4.6 shows the performance.

To further demonstrate the zero-shot learning ability of our framework and the significance on saving human annotation effort, we use the supervised LSTM approach for comparison. The training data of LSTM contains 3,464 sentences with 905 annotated event mentions for the 23 unseen event types. We divide these event annotations into 10 subsets and gradually add one subset (10% of annotations) into the training data of LSTM. Figure 4.4 shows the learning curve. Without any annotated mentions of the 23 unseen test event types in its training set, our transfer learning approach achieves performance comparable to that of the LSTM, which is trained on 3,000 sentences[3] with 500 annotated event mentions.

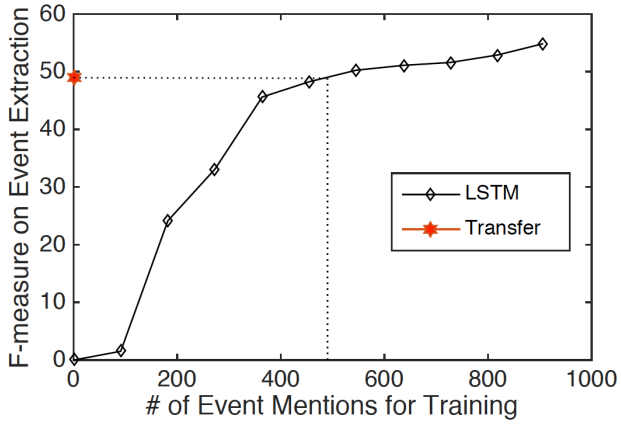

Figure 4.4: Comparison between Our Approach and Supervised LSTM model on 23 Unseen Event Types.

4.6 DISCUSSION: IMPACT OF AMR

We use AMR parsing output to construct event structures. To assess the impact of AMR parser [135] on event extraction, we choose a subset of ERE (Entity, Relation, Event) corpus [154] which has perfect AMR annotations. This subset contains 304 documents

[3]The 3,000 sentences include all the sentences which have not any event annotations.

with 1,022 annotated event mentions of 40 types. We select the top-6 most popular event types (*Arrest-Jail, Execute, Die, Meet, Sentence, Charge-Indict*) with manual annotations of 548 event mentions as seen types. We sample 500 negative event mentions from distinct types of clusters generated from the system [152] based on ERE training sentences. We combine the annotated events for seen types and the negative event mentions, and use 90% for training and 10% for development. For evaluation, we select 200 sentences from the remaining ERE subset, which contains 128 *Attack* event mentions and 40 *Convict* event mentions. Table 4.7 shows the event extraction performances based on perfect AMR and system AMR respectively.

We also compare AMR with Semantic Role Labeling (SRL) output [155] by keeping only the core roles (e.g., *:ARG0, :ARG1*) from AMR annotations. As Table 4.7 shows, compared with SRL, the fine-grained AMR semantic relations such as *:location*, *:instrument* appear to be more informative to infer event argument roles

Method	Trigger Labeling			Argument Labeling		
	P	R	F_1	P	R	F_1
Perfect AMR	79.1	47.1	59.1	25.4	21.4	23.2
Perfect AMR with Core Roles only (SRL)	77.1	47.0	58.4	19.7	16.9	18.2
System AMR	85.7	32.0	46.7	22.6	15.8	18.6

Table 4.7: Impact of AMR and Semantic Roles on Trigger and Argument Extraction (%).

4.7 SUMMARY

Extracting knowledge for a set of new types is usually a costly task. In this chapter, we investigate zero-shot transfer learning, a new learning fashion that can leverage available annotations for a few seen types, e.g., annotated event mentions for Attack, Sentence and Meet, and automatically extract event mentions for all other unseen types (e.g., Convict, Threaten) from a large-scale and extensible target ontology. Without any annotation, our approach can achieve performance comparable to state-of-the-art supervised models trained on about 500 event mentions and 3000 sentences for 23 ACE types. The grounding idea can be applied to a lot of other NLP tasks, e.g., entity recognition, relation extraction, entity linking.

CHAPTER 5: SEMI-SUPERVISED NEW EVENT TYPE INDUCTION AND EVENT DETECTION - AN EXTENSION OF ZERO-SHOT IE

The aforementioned zero-shot learning approach requires a large-scale target event ontology available so that the algorithms can efficiently map each candidate event mention into a particular type. However, in practice, it is usually very expensive and time-consuming to manually craft a large-scale event schema, which defines the types and complex templates of the expected events. So, one further question is, can machines automatically discover a set of new event types and their event mentions by leveraging existing annotations for a few seen types?

5.1 MOTIVATIONS

Recent studies have shown that it's possible to automatically induce an event schema from raw text. Some researchers explore probabilistic generative methods [156, 157, 158, 159] or ad-hoc clustering-based algorithms [160] to discover a set of event types and argument roles. Generally, event schema induction can be divided into two steps: event type induction, aiming to discover a set of new event types for the given scenario, and argument role induction which discovers a set of argument roles for each type. In this chapter, we focus on tackling the first problem only.

We propose a task of semi-supervised event type induction, which aims to leverage available event annotations for a few types, which are called as *seen* types, and automatically discover a set of new *unseen* types, as well as their corresponding event mentions. As a solution, we design a new Semi-Supervised Vector Quantized Variational Autoeocoder framework (short as **SS-VQ-VAE**) which first assigns a discrete latent type representation for each seen and unseen type, and optimizes them during the process of projecting each candidate trigger into a particular seen or unseen type. The candidate triggers are discovered with a heuristic approach. To avoid the type projection to be over-fitted to the set of seen types, we introduce a variational autoencoder (VAE) as a regularizer to enforce the decoder to reconstruct each particular trigger conditioned on its type distribution.

Experiments under the setting of both supervised event detection and new event type induction demonstrate that our approach can not only detect event mentions for seen types with high precision, but also discover high-quality new unseen types.

5.2 APPROACH OVERVIEW

As Figure 5.1 shows, given an input sentence, we first automatically discover all candidate triggers and encode each trigger with a contextual vector using a pre-trained BERT [161] encoder. Then, we predict the type of each candidate trigger by looking up a dictionary of discrete latent representations of all seen and unseen types. Meanwhile, to avoid the type prediction to be over-fitted to seen types, we apply a variational autoencoder as a regularizer to first project each trigger into a latent variational embedding and then reconstruct the trigger conditioned on its type distribution.

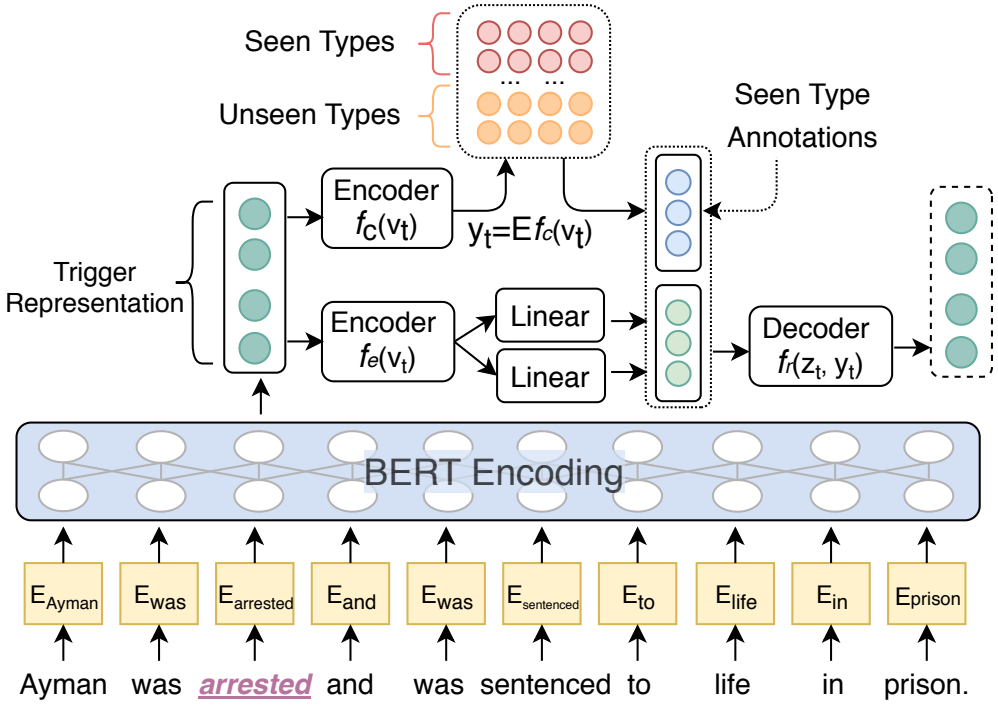

Figure 5.1: Architecture Overview.

5.3 EVENT TRIGGER IDENTIFICATION AND REPRESENTATION LEARNING

Similar to [160], we identify all candidate triggers based on word sense induction. Specifically, for each word, we disambiguate its senses and link each sense to OntoNotes [162] using a word sense disambiguation system — IMS [140] [1]. We consider all noun and verb concepts that can be mapped to OntoNotes senses as candidate triggers. In addition, the concepts that can be matched with verbs or nominal lexical units in FrameNet [163] are also considered as candidate triggers.

[1]We use the OntoNotes based IMS word sense disambiguator (https://github.com/c-amr/camr)

Given a sentence $s = [w_1, ..., w_n]$, where we assume w_i is identified as a candidate trigger, we use a pre-trained BERT encoder to encode the whole sentence and get a contextual representation for w_i. If w_i is split into multiple subwords or a trigger consists of multiple tokens, we use the average of all subword vectors as the final trigger representation.

5.4 EVENT TYPE PREDICTION WITH VECTOR QUANTIZATION

To predict a type for a candidate trigger, an intuitive approach is to learn a classifier using the event annotations of seen types. However, as we also aim to discover a set of unseen types, without any annotations, the classifier for the unseen types cannot be optimized.

To solve this problem, we employ a Vector Quantization [164] strategy. We first define a discrete latent event type embedding space $\boldsymbol{E} \in \mathrm{R}^{k \times d}$, where k is the number of possible event types, and d is the dimensionality of each type embedding $\boldsymbol{e}_i$. Each $\boldsymbol{e}_i$ can be viewed as the *centroid* of the triggers belonging to the corresponding event type. For each seen type, we initialize $\boldsymbol{e}$ with the contextual vector of a trigger which is randomly selected from the corresponding annotations. For each unseen type, we initialize $\boldsymbol{e}$ with the contextual vector of another trigger which is randomly picked from all unannotated event mentions. Assuming there are m seen types, we arbitrarily assign $\boldsymbol{E}^{[1:m]}$ as their type representations.

Given a candidate trigger t and its contextual vector $\boldsymbol{v}_t$, we first apply a linear encoder $f_c(\boldsymbol{v}_t) \in \mathrm{R}^d$ to extract type specific features. Then, we compute a type distribution $\boldsymbol{y}$ based on $f_c(\boldsymbol{v}_t)$ by looking up all the discrete latent event type embeddings with inner-product operation

$$\boldsymbol{y}_t = \boldsymbol{E}^{[1:k]} \cdot f_c(\boldsymbol{v}_t) \tag{5.1}$$

The feature encoder $f_c(.)$ is optimized using all event annotations for seen types (the cross-entropy term in Equation 5.2) and event mentions for unseen types (the second term in Equation 5.2 [2]) as follows

$$\mathcal{L}_c = \sum_{(t,\tilde{y}_t) \in D_s} -\tilde{\boldsymbol{y}}_t \log(\boldsymbol{y}_t) + \sum_{t \in D_u} \max(\boldsymbol{y}_t^{[1:m]}) - \max(\boldsymbol{y}_t^{[m:k]}) \tag{5.2}$$

where $-\tilde{y}_t$ is the ground truth label. D_s and D_u denote the set of annotated event mentions for seen types and new event mentions for unseen types. $\boldsymbol{y}_t^{[1:m]}$ and $\boldsymbol{y}_t^{[m:k]}$ are the type prediction scores for seen and unseen types respectively.

To optimize the type embeddings $\boldsymbol{E}$, we follow the VQ objective [165] and use l_2 error to move the type vector $\boldsymbol{e}_i$ towards the type-specific feature $f_c(\boldsymbol{v}_t)$ (the first term in Equa-

[2] We only apply this term when we know the new event mentions do not belong to any seen types

tion 5.3) while $\boldsymbol{e}_i$ of t is determined by $\boldsymbol{y}_t$. To make sure $f_c(.)$ commits to an embedding, we add a commitment loss (the second term in Equation 5.3)

$$\mathcal{L}_{vq} = ||\text{sg}(f_c(\boldsymbol{v}_t)) - \boldsymbol{e}_i||^2 + ||f_c(\boldsymbol{v}_t) - \text{sg}(\boldsymbol{e}_i)||^2 \tag{5.3}$$

where sg stands for the stop gradient operator, which is defined as identity at forward computation time and has zero partial derivatives, thus effectively constraining its operand to be a non-updated constant.

5.5 VARIATIONAL AUTOENCODER AS REGULARIZER

To avoid the type prediction to be over-fitted to the seen types, we employ a semi-supervised variational autoencoder as a regularizer. The intuition is that each event mention can be generated conditioned on a latent variational embedding $\boldsymbol{z}$ as well as a its corresponding type distribution $\boldsymbol{y}$, which is predicted by the approach described in Section 5.4.

We first describe the semi-supervised variational inference process. It consists of an inference network $q(z|t)$ which is a posterior of the learning of a latent variable z given the trigger t, and a generative network $p(t|z, y)$ to reconstruct the candidate trigger t from the latent variable z and type information y. For each candidate trigger t with human annotated label y, the likelihood $p(t, y)$ can be approximated to a variational lower bound

$$\log p(t, y) \geq \log p(t|y, z) - KL(q(z|t)||p(z)) = -\mathcal{L}(t, y) \tag{5.4}$$

where $\log p(t|z, y)$ is the expectation of reconstruction of t conditioned on z and y, $p(z)$ is the prior Gaussian distribution. For each unlabeled candidate trigger t, the likelihood $p(t)$ approximates to another variational lower bound

$$\log p(t) \geq \sum_y q(y|t)(-\mathcal{L}(t, y)) - q(y|t) \log q(y|t) = -\mathcal{L}(t) \tag{5.5}$$

where $q(y|t)$ is obtained from Equation 5.1.

As for model implementation, given a candidate trigger t and its contextual embedding $\boldsymbol{v}_t$, we first pass it through an encoder $f_e(\boldsymbol{v}_t)$ to extract features. As we assume the latent variatonal embedding $\boldsymbol{z}_t$ follows Gaussian distribution $\boldsymbol{z}_t \sim \boldsymbol{N}(\boldsymbol{\mu}_t, \boldsymbol{\sigma}_t)$, we apply two linear functions to obtain the mean vector $\boldsymbol{\mu}_t = f_\mu(f_e(\boldsymbol{v}_t))$ and a variance vector $\boldsymbol{\sigma}_t = f_\sigma(f_e(\boldsymbol{v}_t))$. For decoding, we employ another linear function to reconstruct $\boldsymbol{v}_t$ from the concatenation of $\boldsymbol{z}_t$ and $\boldsymbol{y}_t$: $\boldsymbol{v}_t^{'} = f_r([\boldsymbol{z}_t : \boldsymbol{y}_t])$. We optimize the following objective for the semi-supervised

VAE

$$\mathcal{L}_v = \sum_{t \in D_u} \mathcal{L}(t) + \sum_{(t,y) \in D_s} \mathcal{L}(t, y) \tag{5.6}$$

The overall loss function for optimizing the whole **SS-VQ-VAE** framework is

$$\mathcal{L} = \alpha \mathcal{L}_c + \beta \mathcal{L}_{vq} + \gamma \mathcal{L}_v \tag{5.7}$$

where α, β and γ are hyper-parameters to balance the three objectives.

5.6 DATA AND EXPERIMENTAL SETUP

We perform experiments on Automatic Content Extraction (ACE) 2005 dataset and evaluate our approach under two settings: (1) supervised event extraction, where the target types include 33 ACE predefined types and *other*. Giving all candidate triggers, the goal is to correctly identify all ACE event mentions and classify them into corresponding types. We follow the same data split with prior work [97, 101, 166] in which 529/30/40 newswire documents are used for training/dev/test set. (2) new event type induction, where only 10 ACE types are seen. Given all ACE annotated event mentions, the goal of this task is to test whether the approach can automatically discover the remaining 23 unseen ACE types and categorize each candidate trigger into a particular seen or unseen type.

In terms of implementation details, we use the pre-trained bert-large-cased [3] model for fine-tuning, and optimize our model with BertAdam. we optimize the parameters with grid search: training epoch 15, learning rate $l \in \{1e-5, 2e-5, 3e-5, 5e-5\}$, gradient accumulation steps $g \in \{1, 2, 3\}$, training batch size $b \in \{5g, 8g, 10g\}$, the hyper-parameters for the overall loss function $\alpha \in \{1.0, 5.0, 10.0\}$, $\beta \in \{0.1, 0.5, 1.0\}$, $\gamma \in \{0.1, 0.5, 1.0\}$. The dimensionality of type embedding as well as latent variational embedding, and the hidden states of $f_c(.)$ are all 500 while the hidden states of $f_e(.)$, $f_\mu(.)$, $f_\sigma(.)$ are all 1024.

We also use several clustering metrics to measure the agreement between the ground truth class assignments and system based unseen type predictions:

Normalized Mutual Info is a normalization of the Mutual Information (MI) score and scales the MI score to be between 0 and 1.

$$NMI(Y, C) = \frac{2 \times I(Y; C)}{[H(Y) + H(C)]} \tag{5.8}$$

[3] https://github.com/google-research/bert

where Y denotes the ground truth class labels, C denotes the cluster labels, $H(.)$ denotes the entropy function and $I(Y;C)$ is the mutual information between Y and C.

Fowlkes Mallows is an evaluation metric to evaluate the similarity between the clusters obtained from our approach and ground-truth labels of the data.

$$FM(Y,C) = \frac{TP}{\sqrt{((TP+FP)\times(TP+FN))}} \tag{5.9}$$

where TP means True Positive, which is calculated as the number of pair of data points which are in the same cluster in Y and in C. FP refers to False Positive, which is calculated as the number of pair of data points which are in the same cluster in Y but not in C. FN is False Negative and calculated as the number of pair of data points which are not in the same cluster in Y but are in the same cluster in C.

Completeness : A clustering result satisfies completeness if all members of a given class are assigned to the same cluster.

$$C(Y,C) = 1 - \frac{H(C|Y)}{H(C)} \tag{5.10}$$

where $H(C|Y)$ is the conditional entropy of the clustering outputs given the class labels and $H(Y)$ is the entropy of the classes.

Homogeneity : A clustering result satisfies completeness if all of its clusters contain only data points which are members of a single class.

$$C(Y,C) = 1 - \frac{H(Y|C)}{H(Y)} \tag{5.11}$$

V Measure is the weighted harmonic mean between homogeneity score and completeness score.

$$V(Y,C) = \frac{(1+\beta)\cdot h\cdot c)}{(\beta\cdot h)+c} \tag{5.12}$$

where h denotes the homogeneity score and c refers to the completeness score.

Methods	Encoder	Trigger Identification			Trigger Detection		
		P	R	F	P	R	F
DMCNN [100]	CNN	80.4	67.7	73.5	75.6	63.6	69.1
JRNN [101]	RNN	68.5	75.7	71.9	66.0	73.0	69.3
JMEE [167]	GCN	80.2	72.1	75.9	76.3	71.3	73.7
Joint3EE [168]	GRU	70.5	74.5	72.5	68.0	71.8	69.8
MOGANED [169]	GAN	-	-	-	79.5	72.3	75.7
BERT-CRF	BERT	73.8	76.9	75.3	70.4	73.3	71.8
DMBERT [170]	BERT	-	-	-	77.6	71.8	74.6
OneIE [171]	BERT	-	-	75.6	-	-	72.8
SS-VQ-VAE w/o VQ-VAE	BERT	78.2	77.8	78.0	73.2	72.9	73.0
SS-VQ-VAE w/o VAE	BERT	80.8	80.2	**80.5**	76.2	75.7	75.9
SS-VQ-VAE	BERT	79.1	81.4	80.2	75.7	77.8	**76.7**

Table 5.1: Supervised Event Detection Performance on ACE 2005 (F-score%).

5.7 SUPERVISED EVENT DETECTION

We compare our approach with several state-of-the-art event extraction methods under the setting of fully supervised event detection, as shown in Table 5.1. We conduct ablation study to testify the impact of the VQ and VAE components: **SS-VQ-VAE w/o VQ-VAE** is only optimized with the classification loss (Equation 5.2) while **SS-VQ-VAE w/o VAE** is optimized with the classification loss (Equation 5.2) and the VQ objective (Equation 5.3).

As we can see, BERT based approaches generally outperform the methods using CNN, RNN or GRU. Our approach achieves the state-of-the-art among all methods. In particular, the recall of our approach is much higher than other methods, which demonstrate the effectiveness of the trigger identification step. It can narrow the learning space of the model. The ablation studies also prove the effectiveness of the VQ and VAE components.

5.8 NEW EVENT TYPE INDUCTION

For new event type induction, we compare our approach with another intuitive baseline, **BERT-C-Kmeans**, which takes in the BERT based trigger representations and group all candidate triggers into clusters with a Constrained K-means [172], a semi-supervised clustering algorithm which enforces all trigger candidates annotated with the same seen type to belong to the same cluster. Table 5.2 shows the performance with several clustering metrics, which measure the agreement between the ground truth class assignments and system based unseen type predictions.

Metrics	C-Kmeans	SS-VQ-VAE
Normalized Mutual Info	8.93	40.88
Adjusted Rand Index	0.15	4.22
Fowlkes Mallows	6.04	31.46
Completeness	8.64	53.57
Homogeneity	9.22	31.19
V Measure	8.92	39.43

Table 5.2: Evaluation of New Event Type Induction on 23 Unseen Types of ACE 2005 (%).

5.9 QUALITATIVE DISCUSSION

As qualitative analysis, we further pick 6 unseen ACE types, including *Trial-Hearing*, *Sentence*, *Marry*, *Demonstrate*, *Convict*, *Merge-Org*, and randomly select at most 100 event mentions for each type. We visualize their type distribution $\boldsymbol{y}$ using TSNE [4]. As Figure 5.2 shows, most of the event mentions that are annotated with the same ACE type tends to be predicted to the same new unseen type.

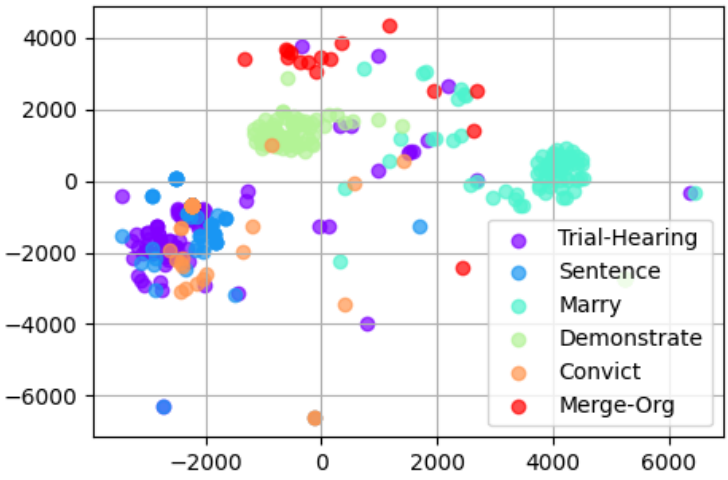

Figure 5.2: Type Distribution of 6 Unseen Types of Event Mentions.

5.10 SUMMARY

In this chapter, we study a more challenging problem of automatically discovering a set of new event types as well as their corresponding event mentions given some event annotations

[4]https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html

for a few event types. It combines the merits of liberal information extraction framework, i.e., high flexibility of the target types, and the merits of zero-shot IE framework, i.e., higher quality by leveraging existing annotations. Experiments show that this approach achieves the state-of-the-art on supervised event extraction and discovers a set of high-quality unseen types. In the future, this framework can be further extended to extract arguments and induce argument roles to discover complete event schemas.

CHAPTER 6: MULTI-LINGUAL COMMON SEMANTIC SPACE CONSTRUCTION: FROM 3 LANGUAGES TO 3000+ LANGUAGES

All the aforementioned IE frameworks, including liberal IE, zero shot IE and the semi-supervised event type induction framework, can extend information extraction tasks to thousands of types without requiring any additional annotated data. However, they are all designed by English. In order to adapt these frameworks to other languages, we describe a general framework to construct a multi-lingual common semantic space, where words from various languages but referring to the same concept share similar semantics.

6.1 MOTIVATIONS

There are about 7,099 known living languages in the world, and more than 3,000 languages have electronic record, e.g., at least a portion of the Christian Bible had been translated into 2,508 different languages.[1] However, the training data for mainstream natural language processing (NLP) tasks such as information extraction and machine translation is only available for dozens of dominant languages. In this paper we aim to construct a multilingual common semantic space where words in multiple languages are mapped into a distributed, language-agnostic semantic continuous space, so that resources and knowledge can be shared across languages.

Words can be clustered through explicit (e.g., sharing affixes of certain linguistic functions) or implicit clues (e.g., sharing neighbors from monolingual word embedding). We hypothesize that the distribution of such clusters should be consistent across multiple languages. We achieve this cluster-level consistency by aligning word clusters across languages. Based on this intuition we propose to create clusters through three kinds of signals as follows, without any extra human annotation effort. Then we aggregate the embedding vectors of words in each cluster and ensure that the clusters (or the words therein) are consistent across multiple languages.

Neighbor based clustering and alignment. We build our common space based on correlational neural network (CorrNet) which is commonly used to learn word representations for multiple views or languages. CorrNet is an extension of autoencoder framework by enabling cross-lingual reconstruction. In contrast to previous work [173, 174], we extend CorrNet to *neighbor-consistent correlation network* by using each word's neighbors (the nearest words within monolingual semantic space) to ensure that the cross-lingual mapping from and to the common semantic space is locally smooth. For instance, the neighboring

[1] https://www.ethnologue.com

words of *China* in English (*Japan*, *India* and *Taiwan*) should be close to the neighboring words of *Cina* in Italian (*Beijing*, *Korea*, *Japan*) in the common semantic space. In other words, we encourage the consistency of neighborhoods across multiple languages.

Character based clustering and alignment. Many related languages share very similar character set, and many words that refer to the same concept share similar compositional characters or patterns, e.g., *China* (English), *Kina* (Danish), and *Cina* (Italian).

Linguistic property based clustering and alignment. Many languages also share linguistic properties, e.g., apposition, conjunction, and plural suffix (English (*-s* / *-es*), Turkish (*-lar* / *-ler*), Somali (*-o*)). Linguists have created a wide variety of linguistic property knowledge bases, which are readily available for thousands of languages. For example, the CLDR (Unicode Common Locale Data Repository) includes closed word classes and affixes indicating various linguistic properties. We propose to take advantage of these language-universal resources to create clusters, where the words within one cluster share the same linguistic property, and build alignment between clusters for common semantic space construction.

We evaluate our approach on monolingual and multilingual QVEC [175] tasks, as well as an extrinsic evaluation on name tagging for low-resource languages. Experiments demonstrate that our framework is effective at capturing linguistic properties and significantly outperforms state-of-the-art multi-lingual embedding learning methods.

6.2 APPROACH OVERVIEW

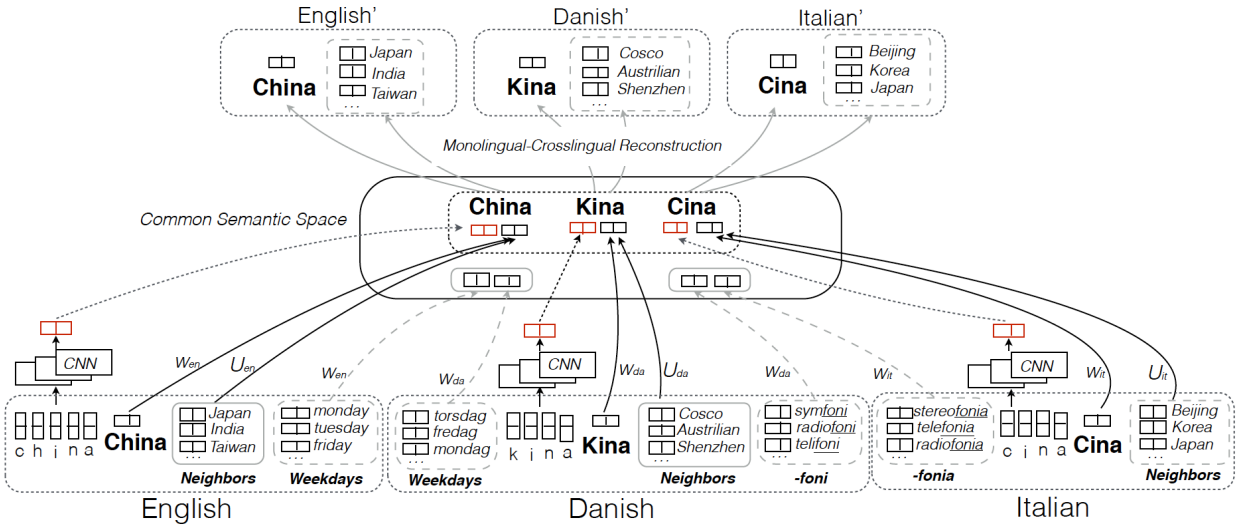

Figure 6.1: Architecture Overview. In each monolingual semantic space, the words within solid rectangle denote a neighbor based cluster and the words within dotted rectangle denote a linguistic property based cluster.

Figure 6.1 shows the overview of our neural architecture.

We project all monolingual word embeddings into a common semantic space based on word-level as well as cluster-level alignments and learn the transformation functions. First, on word-level, we build a neighborhood-consistent CorrNet to augment word representations with neighbor based clusters and align them in the common semantic space. In addition, we apply a language-independent convolutional neural networks to compose character-level word representation and concatenate it with word representation in the common semantic space. Finally, we construct clusters based on linguistic properties, such as closed word classes and affixes, and align them in the common semantic space. We jointly optimize for all the alignments in the common semantic space for each pair of languages.

6.3 BASIC MODEL: CORRNET

We briefly describe the basic model for learning the common semantic space: correlational neural networks (CorrNets) [173, 174]. CorrNets have been widely adopted for learning multilingual or multi-view representations. Figure 6.2 shows the basic architecture of a CorrNet. It combines the advantages of canonical correlation analysis (CCA) and autoencoder (AE).

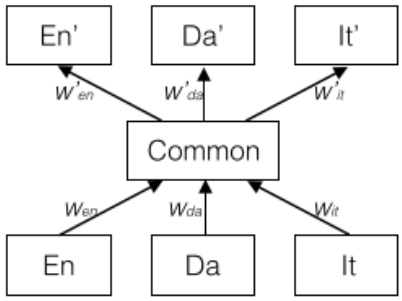

Figure 6.2: CorrNet for Learning Multilingual Word Embeddings

Given the bilingual aligned word pairs between two languages l_1 and l_2, we first use their monolingual word embeddings to initialize each word with a vector and obtain $M_{l_1} \in \mathbb{R}^{|V_{l_1}| \times d_{l_1}}$ and $M_{l_2} \in \mathbb{R}^{|V_{l_2}| \times d_{l_2}}$, where V_{l_1} and V_{l_2} are the bilingual dictionary of l_1 and l_2. $V_{l_1}^i$ is the translation of $V_{l_2}^i$, and d_{l_1} and d_{l_2} are the vector dimensionalities. Then for each language we learn a linear projection function to project M_{l_1} and M_{l_2} into the common semantic space:

$$H_{l_1} = \sigma(M_{l_1} \cdot W_{l_1} + b_{l_1}) \ , \tag{6.1}$$

$$H_{l_2} = \sigma(M_{l_2} \cdot W_{l_2} + b_{l_2}) \ , \tag{6.2}$$

where $H_{l_1} \in \mathbb{R}^{V_{l_1} \times h}$ and $H_{l_2} \in \mathbb{R}^{V_{l_2} \times h}$ are the vector representations for V_{l_1} and V_{l_2} in the common semantic space respectively. h is the vector dimensionality in the shared semantic space. $W_{l_1} \in \mathbb{R}^{V_{l_1} \times h}$ and $W_{l_2} \in \mathbb{R}^{V_{l_2} \times h}$ are the transformation matrices, and b_{l_1} and b_{l_2} are the bias vectors. σ denotes Sigmoid function.

After we project the monolingual embeddings into the common semantic space, we further reconstruct M_{l_1} and M_{l_2} from H_{l_1} and H_{l_2} separately:

$$M_{l_1}^{'} = \sigma(H_{l_1} \cdot W_{l_1}^{\top} + b_{l_1}^{'}) \ , \tag{6.3}$$

$$M_{l_1}^{*} = \sigma(H_{l_2} \cdot W_{l_1}^{\top} + b_{l_1}^{'}) \ , \tag{6.4}$$

$$M_{l_2}^{'} = \sigma(H_{l_2} \cdot W_{l_2}^{\top} + b_{l_2}^{'}) \ , \tag{6.5}$$

$$M_{l_2}^{*} = \sigma(H_{l_1} \cdot W_{l_2}^{\top} + b_{l_2}^{'}) \ , \tag{6.6}$$

where $b_{l_1}^{'}$, $b_{l_2}^{'}$ are the bias vectors. $M_{l_1}^{'}$ and $M_{l_2}^{'}$ are the monolingual reconstructions of M_{l_1} and M_{l_2} from the common space, and $M_{l_1}^{*}$ and $M_{l_2}^{*}$ are cross-lingual reconstructions. $W_{l_1}^{\top}$ and $W_{l_2}^{\top}$ are the transposes of W_{l_1} and W_{l_2} respectively.

To learn the common semantic space, we minimize the distance between the aligned word vectors as well as the loss of monolingual and cross-lingual reconstruction:

$$O_W = \sum_{\{l_i, l_j\} \in A} L(M_{l_i}^{'}, M_{l_i}) + L(M_{l_i}^{*}, M_{l_i}) + L(M_{l_j}^{'}, M_{l_j}) + L(M_{l_j}^{*}, M_{l_j}) + L(H_{l_i}, H_{l_j}) \ , \tag{6.7}$$

where l denotes any specific language that we want to project into the common semantic space, A denotes all bilingual dictionaries, and L denotes a similarity metric. In our work, we use cosine similarity as the similarity metric.

6.4 NEIGHBORHOOD-CONSISTENT CORRNET

CorrNet can project multiple monolingual word embeddings into a common semantic space using bilingual word alignment. However, the same concepts may have different semantic bias in various languages. For example, the top five nearest words of the concept "*China*" are: (*Japan, India, Taiwan, Chinese, Asia*) in English, (*Cosco, Shenzhen, Australian, Shanghai, manufacturing*) in Danish, and (*Beijing, Korea, Japan, aluminum, copper*) in Italian respectively. The neighboring words can reflect the semantic meanings of each concept within each semantic space. In order to ensure the consistency of the neighborhoods within the common semantic space and make the cross-lingual mapping locally smooth, we propose to augment

monolingual word representation with its top-N nearest neighboring words from the original monolingual semantic space.[2]

Given the monolingual embeddings of the bilingual aligned words for two languages l_1 and l_2, M_{l_1} and M_{l_2}, for each word, we extract the top-N nearest neighbors and construct the neighborhood clusters. Each cluster $t_l = \{w_1, w_2, ..., w_{|t_l|}\}$ in language l is represented by

$$c_{t_l} = \frac{1}{|t_l|} \sum_{w \in t_l} E_w \ , \tag{6.8}$$

where E_w denotes the monolingual word embedding for w.

We obtain all the neighborhood cluster vector representations C_{l_1}, C_{l_2} for l_1 and l_2. We incorporate these neighborhood cluster information into the common semantic space when projecting monolingual embeddings:

$$H_{l_1} = \sigma(M_{l_1} \cdot W_{l_1} + C_{l_1} \cdot U_{l_1} + b_{l_1}), \tag{6.9}$$

$$H_{l_2} = \sigma(M_{l_2} \cdot W_{l_2} + C_{l_2} \cdot U_{l_2} + b_{l_2}), \tag{6.10}$$

Besides the monolingual and cross-lingual reconstructions for M_{l_1} and M_{l_2} in CorrNets, we also add monolingual and cross-lingual reconstructions for the neighborhood clusters:

$$C'_{l_1} = \sigma(H_{l_1} \cdot U_{l_1}^\top + b^*_{l_1}) \ , \tag{6.11}$$

$$C^*_{l_1} = \sigma(H_{l_2} \cdot U_{l_1}^\top + b^*_{l_1}) \ , \tag{6.12}$$

$$C'_{l_2} = \sigma(H_{l_2} \cdot U_{l_2}^\top + b^*_{l_2}) \ , \tag{6.13}$$

$$C^*_{l_2} = \sigma(H_{l_1} \cdot U_{l_2}^\top + b^*_{l_2}) \ , \tag{6.14}$$

In addition to optimizing the loss functions described in the Section 6.3, we further optimize the monolingual and cross-lingual reconstruction for neighborhood clusters:

$$O_N \quad = \quad \sum_{\{l_i, l_j\} \in A} L(C'_{l_i}, C_{l_i}) \ + \ L(C^*_{l_i}, C_{l_i}) \ + \ L(C'_{l_j}, C_{l_j}) \ + \ L(C^*_{l_j}, C_{l_j}) \quad , \tag{6.15}$$

6.5 CHARACTER-LEVEL WORD ALIGNMENT

Bilingual word alignment is not always enough to induce a common semantic space, especially for low-resource languages. Although the words that refer to the same concept are

[2]We set N as 5 in our experiments.

not exactly the same in multiple languages, they usually share a set of similar characters, especially in related languages written in the same script, such as Amharic and Tigrinya. For example, the same entity is spelled slightly differently in three languages: *Semsettin Gunaltay* in English, *Şemsettin Günaltay* in Turkish, and *Semsetin Ganoltey* in Somali. Beyond word-level alignment, we introduce character-level alignment by composing word representations from its compositional characters using convolutional neural networks (CNN). For each language, we adopt a language-independent CNN to generate character-level word representation.

Character Lookup Embeddings Let S_l be the character set for language l and $E_{S_l} \in \mathbb{R}^{|S_l| \times d}$ be the character lookup embeddings, where d is the dimensionality of each character embedding. Here, we use a simple yet effective method to induce character embeddings from word embeddings. For each character c, we average the embeddings of all words which contain the character. The character embeddings will be further tuned by the model.

Character-Level CNN [176] The input layer is a sequence of characters of length k for each word. Each character is represented by a d-dimensional lookup embedding. Thus each input sequence is represented as a feature map of dimensionality $d \times k$.

The convolution layer is used to learn the representation for each sliding n-gram characters. We make p_i as the concatenated embeddings of n continuous columns from the input matrix, where n is the filter width. We then apply the convolution weights $W \in \mathbb{R}^{d \times nd}$ to p_i with a biased vector $b \in \mathbb{R}^d$ as follows:

$$p_i^{'} = \tanh(W \cdot p_i + b) \tag{6.16}$$

All n-gram representations $p_i^{'}$ are used to generate the word representation y by max-pooling.

In our experiments, we apply multiple filters with various widths to obtain the representation for word w_i^l. The final character-level word representation $\hat{w}_i^l$ is the concatenation of all word representations with varying filter widths.

Cross-Lingual Mapping Given the bilingual aligned word pairs, we directly minimize the distance of the character-level word representations in the common semantic space by:

$$O_{char} = \sum_{\{l_i, l_j\} \in A} L(\hat{W}_{l_i}^{char}, \hat{W}_{l_j}^{char}) \tag{6.17}$$

The final word representation of w_i^l in the common semantic space is the concatenation of character-level word presentation $\hat{w}_i^l$ and projected word representation h_i^l.

Class Name	Words / Word Pairs
Colors	white, yellow, red, blue, green ...
Weekdays	monday, tuesday, friday, sunday ...
Months	january, february, march, april ...
cardinal numbers	one, two, three, four, five ...
ordinal numbers	first, second, third, fourth, fifth ...
pronouns	i, me, you, he, she, her, they ...
prepositions	of, in, on, for, from, about ...
conjunctions	but, and, so, or, when, while ...
clothes	hat, shirt, pants, skirt, socks ...
-like	(god, godlike), (bird, birdlike) ...
-able	(accept, acceptable), (adopt, adoptable) ...
micro-	(gram, microgram), (chip, microchip) ...
auto-	(maker, automaker), (gas, autogas) ...

Table 6.1: Examples of closed word classes and linguistic properties based clusters

6.6 LINGUISTIC PROPERTY ALIGNMENT

6.6.1 Linguistic Property Alignment

Linguists have made great efforts at building linguistic property knowledge bases for thousands of languages in the world. These knowledge bases include a large number of topological properties (phonological, lexical and grammatical) which we will use to build a high-level alignment between words across languages. We exploit the following resources:

- **CLDR** (Unicode Common Locale Data Repository)[3] which includes multilingual gazetteers for months, weekdays, cardinal and ordinal numbers;
- **Wiktionary**[4] which is a multilingual, web-based collaborative project to create an English content dictionary, includes word and prefix/suffix dictionaries for 1,247 languages;
- **Panlex**[5] database which contains 1.1 billion pairwise translations among 21 million expressions in about 10,000 language varieties.

We mainly exploit two types of linguistic properties to extract word clusters. The first type is closed word classes, such as colors, weekdays, and months. Table 6.1 shows some examples

[3]http://cldr.unicode.org/index/charts
[4]https://en.wiktionary.org
[5]http://panlex.org/

of the word clusters we automatically extracted from CLDR and Wiktionary for English. The second type of word clusters are generated based on morphological information, including affixes that indicate various linguistic functions. These properties tend to be consistent across many languages. For example, "-like" is a suffix denoting "similar to" in English, while in Danish "-agtig" performs the same function. For each affix, we extract a set of word pairs (*basic word*, *extended word with affix*) to denote its semantics from each language.

We extract a set of word clusters from each language, and align the clusters based on their functions defined in CLDR, Wiktionary and Panlex. For each language l, each cluster $r_i^l \in R^l$ contains a set of words or word-pairs sharing the same function. We use the average operation to obtain an overall vector representation for each cluster M_l^R.[6] Then, we project the cluster-level vectors into the shared semantic space and minimize the distance between them:

$$H_{l_i}^R = \sigma(M_{l_i}^R \cdot W_{l_i} + b_{l_i}^R) \ , \tag{6.18}$$

$$H_{l_j}^R = \sigma(M_{l_j}^R \cdot W_{l_j} + b_{l_j}^R) \ , \tag{6.19}$$

$$O_R = \sum_{\{l_i, l_j\} \in A} L(H_{l_i}^R, H_{l_j}^R) \ , \tag{6.20}$$

where W is the same as the W used in Section 6.4 for each language. We finally optimize the sum of the losses by finding the parameters $\theta = \{W_l, b_l, b_l^{'}, U_l, b_l^*, \text{CNN}_l, b_l^R\}$, where l denotes a specific language:

$$O_\theta = O_W + O_N + O_{char} + O_R \tag{6.21}$$

6.7 EXPERIMENTS

6.7.1 Experiment Setup

Previous work [177, 178] evaluated multilingual word embeddings on a series of intrinsic (e.g., monolingual and cross-lingual word similarity, word translation) and extrinsic (e.g., multilingual document classification, multilingual dependency parsing) evaluation tasks. Compared with previous work, we aim at incorporating more linguistic features into the multilingual embeddings, which can be helpful for downstream NLP tasks. In order to evaluate the quality of the multilingual embeddings, we use QVEC [175] tasks (details will be described in Section 6.7.2) as the intrinsic evaluation platform. In addition, to demon-

[6]For each word pair, we use the vector of the extend word minus the vector of the basic word as the vector representation of the word pair.

strate the effectiveness of our common semantic space for knowledge transfer, especially for low-resource scenarios, we adopt the low-resource language name tagging task for extrinsic evaluation.

For fair comparison with state-of-the-art methods on building multi-lingual embeddings [177, 178], we use the same monolingual data and bilingual dictionaries as in their work. We build multilingual word embeddings for 3 languages (*English, Italian, Danish*) and 12 languages (*Bulgarian, Czech, Danish, German, Greek, English, Spanish, Finnish, French, Hungarian, Italian, Swedish*) respectively. The monolingual data for each language is the combination of the Leipzig Corpora Collection[7] and Europarl.[8] The bilingual dictionaries are the same as those used in [177].[9]

For each task, we evaluate the performance of our common semantic space in comparison with previously published multilingual word embeddings (MultiCluster, MultiCCA, Multi-Skip, and MultiCross). MultiCluster [177] groups multilingual words into clusters based on bilingual dictionaries and forces all the words from various languages within one cluster share the same embedding. MultiCCA [177, 179] uses CCA to estimate linear projections for each pair of languages. MultiSkip is an extension of the multilingual skip-gram model [180], which requires parallel data. MultiCross is an approach to unify bilingual word embeddings into a shared semantic space using post hoc linear transformations [178].

Table 6.2 lists the hyper-parameters used in the experiments.

Parameter Name	Value
Monolingual Word Embedding Size	512
Multilingual Word Embedding Size	512
# of Filters in Convolution Layer	20
Filter Widths	1, 2, 3
Batch Size	500
Initial Learning Rate	0.5
Optimizer	Adadelta

Table 6.2: Hyper-Parameters.

6.7.2 Intrinsic Evaluation: QVEC

In order to evaluate the quality of multilingual embeddings, especially on linguistic aspect, we adopt QVEC [175] as the intrinsic evaluation measure. It evaluates the quality of word

[7] http://wortschatz.uni-leipzig.de/en/download/
[8] http://www.statmt.org/europarl/index.html
[9] http://128.2.220.95/multilingual/data/

		3 Languages				12 Languages			
		Monolingual		Multilingual		Monolingual		Multilingual	
		QVEC	QVEC-CCA	QVEC	QVEC-CCA	QVEC	QVEC-CCA	QVEC	QVEC-CCA
MultiCluster		10.8	9.1	63.6	45.8	10.4	9.3	62.7	44.5
MultiCCA		10.8	8.5	63.8	43.9	10.8	8.5	63.9	43.7
MultiSkip		7.8	7.3	57.3	36.2	8.4	7.2	59.1	36.5
MultiCross		-	-	-	-	11.9	8.6	46.4	31.0
CorrNet	W	14.8	11.3	63.6	43.4	14.7	13.2	63.8	43.9
	W+N	**15.9**	12.7	64.5	45.3	15.5	13.6	65.0	46.4
	W+N+Ch	15.2	12.1	66.3	44.5	14.8	12.9	67.2	**47.3**
	W+N+L	15.8	**12.8**	64.3	45.3	**16.3**	**14.5**	65.0	45.9
	W+N+Ch+L	15.5	12.7	**66.5**	**46.3**	14.9	13.1	**67.3**	47.2

Table 6.3: QVEC and QVEC-CCA scores. W: word alignment. N: neighbor based clustering and alignment. Ch: character based clustering and alignment. L: linguistic property based clustering and alignment.

embeddings based on the alignment of distributional word vectors to linguistic feature vectors extracted from manually crafted lexical resources, e.g., SemCor [181].

$$\text{QVEC} = \max_{\sum_j a_{ij} \leq 1} \sum_{i=1}^{D} \sum_{j=1}^{P} r(x_i, s_j) \times a_{ij} \tag{6.22}$$

where $x \in \mathbb{R}^{D \times 1}$ denotes a distributional word vector and $s \in \mathbb{R}^{P \times 1}$ denotes a linguistic word vector. $a_{ij} = 1$ iff x_i is aligned to s_j, otherwise $a_{ij} = 0$. $r(x_i, s_j)$ is the Pearson's correlation between x_i and s_j. QVEC-CCA [177] is extended from QVEC by using CCA to measure the correlation between the distributional matrix and the linguistic vector matrix, instead of cumulative dimension-wise correlation.

Using QVEC and QVEC-CCA, we evaluate the quality of multilingual embeddings for both monolingual (English) and multilingual (English, Danish, Italian) settings.

As shown in Table 6.3, our approaches outperform previous approaches in all cases. Specifically, by augmenting word representation with neighboring words in the common semantic space as in Eq. (6.9), the performance for monolingual QVEC and QVEC-CCA tasks is consistently improved. In addition, by aligning character-level compositional representations and linguistic property based clusters in the shared semantic space, the multilingual representation quality is further improved.

		QVEC		QVEC-CCA	
		Monolingual	Multilingual	Monolingual	Multilingual
40,000	multiCCA	10.8	8.5	63.8	43.9
	CorrNet W	14.8	11.3	63.6	43.4
	CorrNet W+N+Ch+L	**15.5**	**12.7**	**66.5**	**46.3**
10,000	multiCCA	9.8	6.5	63.6	42.3
	CorrNet W	14.8	11.3	63.4	43.0
	CorrNet W+N+Ch+L	**15.4**	**12.1**	**66.4**	**46.2**
2,000	multiCCA	9.9	6.2	63.6	40.9
	CorrNet W	14.5	7.1	62.0	39.2
	CorrNet W+N+Ch+L	**14.7**	**11.7**	**66.6**	**45.5**

Table 6.4: Results using bilingual lexicons with varying sizes (40,000, 10,000, 2,000) and three languages. CorrNet W+N+Ch+L is the proposed approach with all the cluster types.

6.7.3 Impact of Bilingual Dictionary Size

In order to show the impact of the size of bilingual lexicons, we use three languages as a case study, and gradually reduce the size of the lexicons for each pair of languages from 40,000 to 10,000 and to 2,000. Both MultiCluster and MultiSkip by default take advantage of identical strings from any pair of languages when they learn the multilingual embeddings. For fair comparison, we thus use MultiCCA as a baseline. Table 6.4 shows the results. We observe that both MultiCCA and CorrNet approaches are sensitive to the size of the bilingual lexicons. Our approach on the other hand could maintain high performance, even when the bilingual lexicons were reduced to 2,000.

6.7.4 Low-Resource Name Tagging

We evaluate the quality of multilingual embeddings on a downstream task by using the embeddings as input features. Here, we use low-resource language name tagging as a target task. We experiment with two sets of languages. The first set Amh+Tig consists of Amharic and Tigrinya. Both languages share the same Ge'ez script and descend from the proto-Semitic language family. The other set Eng+Uig+Tur consists of one high-resource language (English), one medium-resource language (Turkish) and one low-resource language (Uighur). It also consists of two distinct language scripts: English and Turkish use Latin script while Uighur uses Arabic script.

We use LSTM-CRF architecture [40, 41, 182] for name tagging. Table 6.5 shows the

	Amh	Tig	Uig	Tur	Eng
Train	1,506	1,585	1,711	3,404	14,029
Dev	167	176	190	378	3,250
Test	711	440	476	1,604	3,453

Table 6.5: Data statistics (# of Sentences) for name tagging

		Multilingual		
	Mono-lingual		CorrNet	
		MultiCCA	W	W+N+Ch+L
Amh	52.0	50.6	52.4	**55.8**
Tig	78.2	**78.4**	77.9	77.6
Uig	**70.0**	63.6	66.8	66.0
Tur	73.9	65.3	72.4	**75.6**

Table 6.6: Name tagging result (F-score, %) using monolingual embedding and multilingual embeddings.

statistics of training, development, and test sets for each language released by Linguistic Data Consortium (LDC).[10] For each language pair in each language set, we combine the bilingual aligned words extracted from Wiktionary and extracted from monolingual dictionaries based on identical strings.[11] We evaluate the quality from several aspects:

Monolingual embedding quality evaluation Table 6.6 shows the name tagging performance for each language using the original monolingual embeddings and multilingual embeddings. For both Amharic and Turkish, the multilingual embeddings learned from our approach significantly improve over the monolingual embeddings, compared to MultiCCA. In the case of Uighur, all the multilingual embeddings fail to outperform the original monolingual embeddings. We conjecture that this is due to the use of Arabic script in Uighur, which differs from Turkish and English.

Cross-Lingual Direct Transfer We further demonstrate the effectiveness of our multilingual embeddings on direct knowledge transfer. In this setting, we train a name tagger on one or two languages using multilingual embeddings and test it on a new language without any annotated data. Table 6.7 shows the performance. For each testing language, our

[10]The annotations are from: Amh (LDC2016E87), Tig (LDC2017E27), Uig (LDC2016E70), Tur (LDC2014E115), Eng [36]

[11]We extracted 23,781 pairs of words for Amh and Tig, 16,868 pairs for Eng and Tur, 3,353 pairs for Eng and Uig, and 3,563 pairs for Tur and Uig.

Train	Test	MultiCCA	CorrNet W	CorrNet W+N+Ch+L
Amh	Tig	15.5	28.3	**31.7**
Tig	Amh	11.1	12.8	**23.3**
Eng	Uig	8.4	**16.9**	15.4
Tur	Uig	1.1	18.1	**25.6**
Eng+Tur	Uig	8.0	20.3	**20.6**
Eng	Tur	20.6	**21.4**	17.3
Uig	Tur	10.4	10.1	**17.7**
Eng+Uig	Tur	18.5	21.1	**29.4**

Table 6.7: Name tagging performance (F-score, %) when the tagger was trained on a source language and tested on a target language. CorrNet W+N+Ch+L is the proposed approach with all the cluster types.

approach achieves better performance than MultiCCA and CorrNet. The closer that the languages are, such as Amharic and Tigrinya, and Turkish and Uighur, the better performance could be achieved, even when they may have distinct language scripts (e.g., Turkish and Uighur).

We also notice that a larger extra annotation from another language does not necessarily result in the improvement. For instance, the proposed approach (CorrNet W+N+Ch+L) suffers from English annotated examples when tested on Turkish. This suggests that we need to be careful and aware of linguistic properties among different languages for transfer learning.

Mutual enhancement We finally show the improvement by adding more cross-lingual annotated data and using multilingual embeddings in Table 6.8. The multilingual embeddings learned by our approach consistently outperforms MultiCCA. More specifically, when there are not enough annotated examples, the performance could be improved by incorporating annotated examples from other languages. This is evident for Amharic, Tigrinya and Uighur.

6.8 SUMMARY

In this chapter, we investigate a general framework to effectively transfer knowledge and resources across various languages, that is, constructing a common semantic space for multiple languages based on a cluster-consistent correlational neural network. It combines word-level

Train	Test	MultiCCA	CorrNet W	CorrNet W+N+Ch+L
Tig+Amh	Amh	52.9	52.1	**56.5**
Amh+Tig	Tig	78.0	**78.1**	**78.7**
Eng+Uig	Uig	67.9	67.8	**68.3**
Tur+Uig	Uig	67.7	67.5	**68.8**
Eng+Tur+Uig	Uig	**68.7**	67.4	65.9
Uig-Tur	Tur	65.9	69.2	**72.8**
Eng-Tur	Tur	66.9	70.4	**73.4**
Eng+Uig+Tur	Tur	67.5	68.5	**72.9**

Table 6.8: Name tagging performance (F-score, %) when the training set for the tagger was enhanced by annotated examples in other languages. CorrNet W+N+Ch+L is the proposed approach with all the cluster types.

alignment and multi-level cluster alignment, including neighbor based clusters, character-level compositional word representations, and linguistic property based clusters induced from the readily available language-universal linguistic knowledge bases. This approach achieved consistently higher correlation on QVEC tasks than state-of-the-art multilingual embedding learning methods, and achieved up to 24.5% absolute F-score gain over the state of the art on low-resource language name tagging task.

In the future, this framework can be further extended to multi-lingual and multi-media, where both words, entities, events from natural language text and images and videos are all represented within a unified semantic space, which can serve as bridge for cross-media knowledge transfer.

CHAPTER 7: WHAT IF THERE IS NO BILINGUAL LEXICON AVAILABLE: CROSS-LINGUAL ADVERSARIAL TRANSFER

The multilingual common semantic space can serve as a bridge to transfer all available resources from resource-dominant languages to low-resource languages, however, it still requires a small size of bilingual lexicon for each pair of languages, which may still be difficult to get. In this chapter, we investigate adversarial training and discuss how to automatically learn language-agnostic features without using any bilingual alignment signals.

7.1 MOTIVATIONS

Low-resource language name tagging is an important but challenging task. An effective solution is to perform cross-lingual transfer, by leveraging the annotations from high-resource languages. Most of these efforts achieve cross-lingual annotation projection based on bilingual parallel corpora combining with automatic word alignment [183, 184, 185, 186, 187], bilingual gazetteers [18, 188], cross-lingual word embedding [19, 189, 190], or cross-lingual Wikification [17, 20, 191, 192], but these resources are still only available for dozens of languages. Recent efforts on multi-task learning model each language as one single task while all the tasks share the same encoding layer [193, 194, 195]. These methods can transfer knowledge via the shared encoder without using bilingual resources. However, different languages usually have different underlying sequence structures, as shown in Figure 7.1. Without an explicit constraint, the encoder is not guaranteed to extract language-independent sequential features. Moreover, when the size of annotated resources is not balanced, the encoder is likely to be biased toward the resource-dominant language.

NED: Sedert het begin[1] van de Europese integratie[2] is het mededingingsbeleid[3] van groot belang[4] voor de Europese Unie[5].

ENG: The European Union[5]' s competition policy[3] has been of central importance[4] since European integration[2] began[1].

ESP: La política de competencia[3] de la Unión Europea[5] ha sido de central importancia[4] desde que se inició[1] la integración europea[2].

Figure 7.1: Example of parallel sentences between English (ENG), Spanish (ESP) and Dutch (NED) from Europarl Parallel Corpus [196]. The information units with the same color and superscript are aligned.

Considering these challenges, we develop a new neural architecture which can effectively transfer resources from source languages to improve target language name tagging. Our neural architecture is built upon a state-of-the-art sequence tagger: bi-directional long short-term memory as input to conditional random fields (Bi-LSTM-CRF) [40, 41, 182], integrated with multi-level adversarial transfer: (1) word level adversarial transfer, similar to [61], applying a projection function on the source language and a discriminator to distinguish each word of the target language from that of the source language, resulting in a bilingual shared semantic space; (2) sentence-level adversarial transfer, where a discriminator is trained to distinguish each sentence of the target language from that of the source language,[1] and a sequence encoder is applied to each sentence of both languages to prevent the discriminator from correctly predicting the source of each sentence, yielding language-agnostic sequential features. These features can better facilitate the resource transfer from the source language to the target language.

7.2 APPROACH OVERVIEW

Figure 7.2 shows the overview of our neural architecture. It consists of three components:

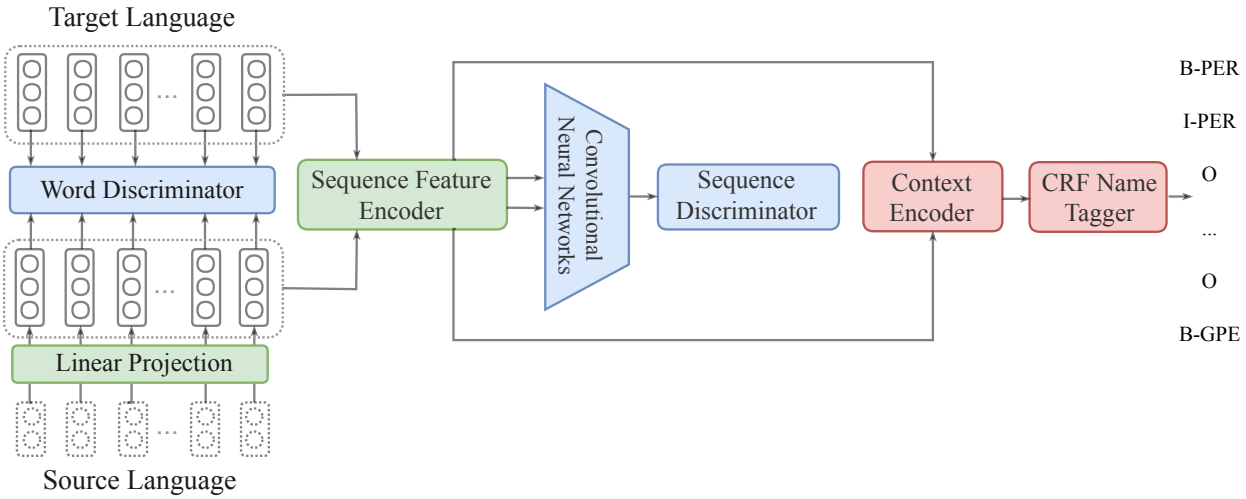

Figure 7.2: Architecture overview.

Cross-lingual word embedding learning with adversarial training: Given pre-trained monolingual word embeddings for a target language t and a source language s, we first apply a mapping function to each word representation from s, then feed both the

[1]For the name tagging task, 'sequence' always means 'sentence.'

projected source word representations and the target word representations to a word discriminator to predict the language of each word. If the discriminator cannot distinguish the language of t from the projection of s, then we consider t and the projection of s to be in a shared space.

Language-agnostic sequential feature extraction: For each sentence of t and s, we apply a sequence encoder to extract sequential features, and a Convolutional Neural Network (CNN) [197] based sequence discriminator to predict the language source of each sentence. The sequence encoder is trained to prevent the sequence discriminator from correctly predicting the language of each sentence, such that it finally extracts language-agnostic sequential features.

Language-independent name tagger The language-agnostic sequential features from both t and s are further fed into a context encoder to better capture and refine contextual information and a conditional random field (CRF) [82] based name tagger.

Next we show the details of each component in our architecture.

7.3 WORD-LEVEL ADVERSARIAL TRANSFER

To better leverage the resources from the source language, our first step is to construct a shared semantic space where the words from the source and target languages are semantically aligned. Without requiring any bilingual gazetteers, recent efforts [61, 198, 199] explore unsupervised approaches to learn cross-lingual word embeddings and achieve comparable performance to supervised methods. Following these studies, we perform word-level adversarial training to automatically align word representations from s and t.

Assume we are given pre-trained monolingual word embeddings $\mathbf{V}_t = \{\mathbf{v}_1^t, \mathbf{v}_2^t, ..., \mathbf{v}_N^t\} \in \mathbb{R}^{N \times d_t}$ for t, and $\mathbf{V}_s = \{\mathbf{v}_1^s, \mathbf{v}_2^s, ..., \mathbf{v}_M^s\} \in \mathbb{R}^{M \times d_s}$ for s, where $\mathbf{v}_i^t$ and $\mathbf{v}_j^s$ are the vector representations of words w_i^t and w_i^s from t and s, N and M denote the vocabulary sizes, d_t and d_s denote the embedding dimensionality of t and s respectively. We then apply a mapping function f to project s into the same semantic space as t:

$$\widetilde{\mathbf{V}}_s = f(\mathbf{V}_s) = \mathbf{V}_s\mathbf{U} \tag{7.1}$$

where $\mathbf{U} \in \mathbb{R}^{d_s \times d_t}$ is the transformation matrix. $\widetilde{\mathbf{V}}_s \in \mathbb{R}^{M \times d_t}$ are the projected word embeddings for s, and $\Theta_f = \{\theta_f\}$ denotes the set of parameters to be optimized for f.

Similar to [200], [61], and [199], we constrain the transformation matrix $\mathbf{U}$ to be orthogonal with singular value decomposition (SVD) to reduce the parameter search space:

$$\mathbf{U} = \mathbf{A}\mathbf{B}^\top \text{ , with } \mathbf{A}\Sigma\mathbf{B}^\top = \text{SVD}(\widetilde{\mathbf{V}}_s\mathbf{V}_s^\top) \tag{7.2}$$

To automatically optimize the mapping function f without using extra bilingual signals, we introduce a multi-layer perceptron D as a word discriminator, which takes word embeddings of t and projected word embeddings of s as input features and outputs a single scalar. $D(w_i^*)$ represents the probability of w_i^* coming from t. The word discriminator is trained by minimizing the binary cross-entropy loss:

$$L_{dis}^w = -\frac{1}{I_{t;s}} \cdot \sum_{i=0}^{I_{t;s}} \Big(y_i \cdot \log(D(w_i^*)) + (1 - y_i) \cdot \log(1 - D(w_i^*)) \Big) \text{ , } y_i = \delta_i(1 - 2\epsilon) + \epsilon \text{ , } \tag{7.3}$$

where $\delta_i = 1$ when w_i^* is from t and $\delta_i = 0$ otherwise. $I_{t;s}$ represents the number of words sampled from the vocabulary of t and s together. ϵ is a smoothed value added to the positive and negative labels. $\Theta_{dis} = \{\theta_D\}$ is the parameter set.

The mapping function f and word discriminator D are two adversarial players, thus we flip the word labels and optimize f by minimizing the following loss:

$$L_f^w = -\frac{1}{I_{t;s}} \cdot \sum_{i=0}^{I_{t;s}} \Big((1 - y_i) \cdot \log(D(w_i^*)) + y_i \cdot \log(1 - D(w_i^*)) \Big) \text{ , } y_i = \delta_i(1 - 2\epsilon) + \epsilon \tag{7.4}$$

Following the standard training procedures of deep adversarial networks [201], we train the word discriminator and the mapping function successively with stochastic gradient descent (SGD) [202] to minimize L_{dis}^w and L_f^w. Similar to [61], after word-level adversarial training, we also adopt a refinement step to construct a bilingual dictionary for the top-k most frequent words in the source language[2] based on $\widetilde{\mathbf{V}}_s$ and $\mathbf{V}_t$, and further optimize $\mathbf{U}$ with Equation 7.2 in a supervised way.

7.4 SENTENCE-LEVEL ADVERSARIAL TRANSFER

Once s is projected into the same semantic space as t, we can regard both sentences as coming from one unified language and directly project annotations from s to t. However, name tagging not only relies on word level features, but also on sequential contextual features for entity type classification. Without constraints, the sequence encoder can only extract

[2]We set k=15,000 in our experiment.

sequential features for both t and s based on their final training signals while these features are not necessarily beneficial to the target language. Thus, we further design sentence level adversarial transfer to encourage the encoder to extract language-agnostic sequential features.

Given a sentence $x^t = \{w_1^t, w_2^t, ...\}$ from t and a sentence $x^s = \{w_1^s, w_2^s, ...\}$ from s, we first use $\mathbf{V}_t$ and $\widetilde{\mathbf{V}}_s$ to initialize a vector representation for each w_i^t and w_i^s. We also apply a character-based CNN (denoted as CharCNN) [176] for each language to compose a word representation from its characters. For each word, we concatenate its word representation and character based representation. Then we feed the sequence of vector representations into a weight sharing Bi-LSTM encoder E to obtain sequential features $\mathbf{H}_t = \{\mathbf{h}_1^t, \mathbf{h}_2^t, ...\}$ and $\mathbf{H}_s = \{\mathbf{h}_1^s, \mathbf{h}_2^s, ...\}$ for x^t and x^s respectively. The parameter set of optimizing both language-dependent CharCNN and the sequence encoder can be denoted as $\Theta_e = \{\theta_{\text{CharCNN}_t}, \theta_{\text{CharCNN}_s}, \theta_E\}$.

Based on these sequential features, we use a sequence discriminator to predict the language source of each sentence. Given a sentence x^* and its sequential features $\mathbf{H} = \{\mathbf{h}_1^*, \mathbf{h}_2^*, ...\}$ from E, we first apply a language-independent CNN with max-pooling to get an overall vector representation for x^*, then feed it into another multi-layer perceptron, $\tilde{D}$, to predict the probability that x^* comes from language t. The sequence discriminator is trained by minimizing the following binary cross-entropy loss:

$$L_{dis}^x = -\frac{1}{\tilde{I}_{t;s}} \cdot \sum_{i=0}^{\tilde{I}_{t;s}} \left(\tilde{y}_i \cdot \log(\tilde{D}(x_i^*)) + (1 - \tilde{y}_i) \cdot \log(1 - \tilde{D}(x_i^*)) \right), \tilde{y}_i = \tilde{\delta}_i(1 - 2\eta) + \eta \ , \quad (7.5)$$

where $\tilde{\delta}_i = 1$ if the sentence x_i^* is from t and $\tilde{\delta}_i = 0$ otherwise. $\tilde{I}_{t;s}$ represents the number of sentences sampled from the whole data set of t and s. η is another smoothed value for sequence labels. $\Theta_{\widetilde{dis}} = \{\theta_{\text{CNN}}, \theta_{\tilde{D}}\}$ denotes the parameter set for optimizing the sequence discriminator.

The sequence encoder E and the sequence discriminator $\tilde{D}$ are two adversarial players and E is optimized by trying to fool $\tilde{D}$ to correctly predict the language source of each sentence. Thus we flip the sequence labels and optimize E by minimizing the following loss:

$$L_e^x = -\frac{1}{\tilde{I}_{t;s}} \cdot \sum_{i=0}^{\tilde{I}_{t;s}} \left((1 - \tilde{y}_i) \cdot \log(\tilde{D}(x_i^*)) + \tilde{y}_i \cdot \log(1 - \tilde{D}(x_i^*)) \right), \tilde{y}_i = \tilde{\delta}_i(1 - 2\eta) + \eta \quad (7.6)$$

7.5 NAME TAGGER TRAINING

With the language-agnostic sequential features from E, we can directly combine all annotated training data from both t and s to train the name tagger for t. To do so, we feed the sequential features from E to another Bi-LSTM encoder E_c to refine the context information for each token, and use a CRF output layer to render predictions for each token, which can effectively capture dependencies among name tags (e.g., an "inside-organization" token cannot follow a "beginning-person" token).

Algorithm 7.1 Multi-Level Adversarial Training for Improving Target Language Name Tagging

Input: Monolingual pre-trained word embeddings $\mathbf{V}_t$ for target language t, and $\mathbf{V}_s$ for source language s. Annotated sentence set Δ_t for t and Δ_s for related language s.

1. **for** *iter* = 1 to *word_epoch* **do**
2. **for** $a = 1$ to *word_dis_steps* **do**
3. sample a batch of words $\mathbf{b}_t \sim \mathbf{V}_t$, $\mathbf{b}_s \sim \mathbf{V}_s$
4. $loss = L^w_{dis}([\mathbf{b}_t, f(\mathbf{b}_s)])$
5. update Θ_{dis} to minimize $loss$
6. sample a batch of words $\mathbf{b}_t^{'} \sim \mathbf{V}_t$, $\mathbf{b}_s^{'} \sim \mathbf{V}_s$
7. $loss^{'} = L^w_f([\mathbf{b}_t^{'}, f(\mathbf{b}_s^{'})])$
8. update Θ_f to minimize $loss^{'}$
9. build a parallel dictionary with $\mathbf{V}_t$ and $f(\mathbf{V}_s)$ and refine projected word embeddings $\widetilde{\mathbf{V}}_s = f(V_s)$
10. **for** *iter* = 1 to *seq_epoch* **do**
11. sample a batch of sentences $\tilde{\mathbf{b}}_t \sim \Delta_t$, $\tilde{\mathbf{b}}_s \sim \Delta_s$
12. extract sequential features from $\tilde{\mathbf{b}}_t$, $\tilde{\mathbf{b}}_s$ with E
13. $loss = L^x_{dis}([E(\tilde{\mathbf{b}}_t), E(\tilde{\mathbf{b}}_s)])$
14. update Θ_e, $\Theta_{\widetilde{dis}}$ to minimize $loss$
15. **for** $g = 1$ to *seq_tagger_steps* do
16. sample a batch of sequences $\tilde{\mathbf{b}}_t^{'} \sim \Delta_t$, $\tilde{\mathbf{b}}_s^{'} \sim \Delta_s$
17. $loss^{'} = L^x_e([E(\tilde{\mathbf{b}}_t^{'}), E(\tilde{\mathbf{b}}_s^{'})]) + L_{crf}([\tilde{\mathbf{b}}_t^{'}, \tilde{\mathbf{b}}_s^{'}])$
18. update Θ_e, Θ_c to minimize $loss^{'}$

Specifically, given an input sentence $x = \{w_1, w_2, ...w_n\}$, we extract language-agnostic sequential features with E, and further obtain a new sequence of contextual features $\widetilde{\mathbf{H}} = \{\tilde{\mathbf{h}}_1, \tilde{\mathbf{h}}_2, ..., \tilde{\mathbf{h}}_n\}$ with E_c. Then we a apply a linear layer ℓ to further convert each $\tilde{\mathbf{h}}_i$ to a score

vector $\mathbf{y}_i$, in which each dimension denotes the predicted score for a tag (the starting, inside or outside of a name mention with a pre-defined entity type). Then we feed the sequence of score vectors $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_n\}$ into the CRF layer. The score of a sequence of tags $\mathbf{Z} = \{z_1, z_2, ..., z_n\}$ is defined as:

$$Score(x, \mathbf{Y}, \mathbf{Z}) = \sum_{i=1}^{n} (R_{z_{i-1}, z_i} + Y_{i, z_i}) \tag{7.7}$$

where R is a transition matrix and $R_{p,q}$ denotes the binary score of transitioning from tag p to tag q. $Y_{i,z}$ represents the unary score of assigning tag z to the i-th word.

Given the annotated sequence of tags $\mathbf{Z}$, the CRF loss is:

$$L_{crf} = \log \sum_{\mathbf{Z}' \in \tilde{\mathbf{Z}}} e^{Score(x, \mathbf{Y}, \mathbf{Z}')} - Score(x, \mathbf{Y}, \mathbf{Z}) \tag{7.8}$$

where $\tilde{\mathbf{Z}}$ is the set of all possible tagging paths. The parameter set for optimizing the name tagger can be denoted as $\Theta_c = \{\theta_{E_c}, \theta_\ell, \theta_{\text{CRF}}\}$.

We jointly optimize the sequence encoder E, the context encoder E_c and the CRF together by minimizing the loss $L^{'} = L_e^x + L_{crf}$, and successively minimize L_{dis}^x and $L^{'}$ with SGD. The end-to-end training for our neural architecture is described in Algorithm 7.1.

7.6 DATA AND EXPERIMENTAL SETUP

7.6.1 Dataset

We evaluate our methods from multiple settings. We first evaluate our architecture on 10 low-resource languages from the DARPA LORELEI project. The annotations are released by the Linguistic Data Consortium (LDC).[3] Each dataset has four predefined name types: person (PER), organization (ORG), location (LOC) and geo-political entity (GPE). For each target low-resource language, we choose a source language if they are from the same language family or use the same script. To show the impact of resource transfer between distinct languages, we also use English as a source language for each target low-resource language. We create the English annotated resource by combining the TAC-KBP 2015 English Entity Discovery and Linking [203] data set and the Automatic Content Extraction

[3]The annotations are from: am (LDC2016E87), ti (LDC2017E39), ar (LDC2016E89), fa (LDC2016E93), om (LDC2017E27), so (LDC2016E91), sw (LDC2017E64), yo (LDC2016E105), ug (LDC2016E70), uz (LDC2016E29)

(ACE2005) data set.[4] To avoid the impact of parameter initialization, we perform 5-fold cross validation. For each experiment, we run twice and get the averaged F-score. Table 7.1 shows the statistics of each data set.

Language	# of Sents	# of Tokens	# of Names
Amharic (am)	4,770	71,399	3,891
Tigrinya (ti)	5,023	95,364	6,201
Arabic (ar)	4,781	80,715	4,937
Farsi (fa)	3,855	72,629	3,966
Oromo (om)	2,987	52,876	4,985
Somali (so)	3,453	78,400	5,571
Swahili (sw)	4,155	96,902	6,044
Yoruba (yo)	1,599	46,084	2,016
Uyghur (ug)	3,961	60,999	2,575
Uzbek (uz)	11,135	177,816	10,937
English (en)	17,936	388,120	23,938

Table 7.1: Data set statistics for each low-resource language.

We also evaluate our approach on high-resource languages. We use Dutch (nl) and Spanish (es) data sets from the CoNLL 2002 [204] shared task as target languages, and use English (en) data from the CoNLL 2003 [36] shared task as the source language. All the data sets have four pre-defined name types: PER, ORG, LOC and miscellaneous (MISC). Table 7.2 shows the statistics of these data sets.

Language	Resource	Train	Dev	Test
English (en)	source language	204,567 (23,499)	51,578 (5,942)	46,666 (5,648)
Dutch (nl)	target language	202,931 (13,344)	37,761 (2,616)	68,994 (3,941)
Spanish (es)	target language	264,715 (18,797)	52,923 (4,351)	51,533 (3,558)

Table 7.2: CoNLL data set statistics: # of tokens and # of names (between parentheses).

For fair comparison, we use the same pre-trained word embeddings of English, Dutch and Spanish as [195], while for each low-resource language we train their word embeddings using the documents from their LDC packages with FastText.[5] Table 7.3 lists the key hyper-parameters we used in our experiments.

[4]The data sets are LDC2015E103 and LDC2006T06

[5]`https://fasttext.cc/`

Parameter Name	Value
Monolingual Embedding Size	100
CharCNN Filter Size	25
CharCNN Filter Widths	[2, 3]
LSTM Hidden Size	100
Droupout Rate	0.5
Smoothing Value ϵ for Word Discriminator	0.1
Word Adversarial Training Epochs	5
Smoothing Value η for Sequence Discriminator	0.3
Sequence Adversarial & Name Tagging Training Epochs	60
# of Steps for Sequence Tagging Training	5
Batch Size	20
Initial Learning Rate	0.01
Optimizer	SGD

Table 7.3: Hyper-parameters.

7.6.2 Baselines

We compare our methods with three categories of baseline methods:[6]

- **Monolingual Name Tagging** Using monolingual annotations only, the current state-of-the-art name tagging model is the Bi-LSTM-CRF network [40, 41, 182].[7]

- **Multi-Task Learning** [195] apply multi-task learning to boost name tagging performance by introducing additional annotations from source languages using a weight sharing context encoder across multiple languages.

- **Language Universal Representations** We apply word adversarial transfer only to project the source language into the same semantic space as the target language, then train the name tagger on the annotations of source and target languages. Word-Adv^1 refers to the approach which is directly trained on the combination of the annotations, while Word-Adv^2 refers to the baseline that is first trained on the target language annotations and then further tuned on the related language annotations.

[6] All the baselines are trained for 100 epochs

[7] For each word, we also combine its word embedding with a CharCNN based representation.

7.7 CROSS-LINGUAL TRANSFER WITH ZERO TARGET LANGUAGE ANNOTATED RESOURCE

We first evaluate our approach on a cross-lingual transfer setting without using any annotated training data from the target language. We conduct experiments on 8 low-resource languages. Among those, some pairs, such as Amharic (am) and Tigrinya (ti), Oromo (om) and Somali (so), or Yoruba (yo) and Swahili (sw), are from the same language family and are closely related, while some are not, such as Arabic (ar) and Farsi (fa). Since our approach requires some unlabeled sentences from the target language to train the sentence-level discriminator, we entirely remove the annotations from the annotated data set of the target language. Table 7.4 presents the results.

target (source)	Cross-Lingual Word-Adv[1]	Multitask Learning	Our Approach
am (ti)	15.19	19.72	**26.86**
ti (am)	16.20	9.06	**29.36**
ar (fa)	1.53	3.52	**13.83**
fa (ar)	2.59	0.91	**11.14**
om (so)	4.66	3.40	**14.14**
so (om)	4.12	2.98	**20.02**
sw (yo)	7.20	5.60	**18.25**
yo (sw)	13.07	6.14	**23.73**

Table 7.4: Cross-lingual transfer when the target language has no resources (F-score %).

Our approach significantly outperforms the previous methods on all languages. Specifically, compared with the Word-Adv[1] baseline, which only performs word-level adversarial transfer, our approach achieves 10% absolute F-score gain on average, which demonstrates the effectiveness of the sentence-level adversarial transfer. In addition, compared with [195], who only apply a shared context-encoder to transfer the knowledge, our approach not only includes a language-sharing encoder, but also performs multi-level adversarial training to encourage the semantic alignment of words from both languages and a sequence encoder to extract language-agnostic sequential features.

7.8 CROSS-LINGUAL TRANSFER FOR LOW-RESOURCE LANGUAGES

We also investigate the impact of cross-lingual transfer when the target languages have some annotated resources. For each target low-resource language, we explore the use of a related low-resource language vs. using the high-resource English as our source language. Table 7.5 shows the performance on 10 low-resource languages.

Comparing cross-lingual embedding based baselines to the monolingual baseline, we observe that for most low-resource languages, directly adding the annotations from the source language to the target language slightly hurts the model. This suggests that when the training data for the target language is not enough, the model will be very sensitive to noise. The multitask learning based baseline [195] performs better than the monolingual baseline only when the target and source languages are very close, such as Amharic (am) and Tigrinya (ti), or Swahili (sw) and Yoruba (yo).

target (related)	Monolingual Bi-LSTM-CRF	Cross-Lingual Embedding Word-Adv[1]	Word-Adv[2]	Multitask Learning	Our Approach Multi-Adversarial
am (ti)	72.23	72.15	72.01	72.35	**73.98**
ti (am)	74.68	74.43	74.83	74.71	**74.93**
ar (fa)	48.92	48.37	47.90	47.53	**49.76**
fa (ar)	64.35	63.93	64.43	63.21	**65.09**
om (so)	76.37	76.43	76.19	76.18	**77.19**
so (om)	77.63	77.31	77.13	77.99	**78.15**
sw (yo)	77.01	77.31	77.85	**77.86**	76.28
yo (sw)	68.97	68.89	69.62	70.12	**70.59**
ug (uz)	68.73	68.53	68.29	68.39	**69.46**
uz (ug)	74.59	74.21	74.74	74.56	**75.37**
am (en)	72.23	72.43	71.63	72.22	**73.35**
ti (en)	74.68	74.61	74.69	74.68	**74.80**
ar (en)	48.92	48.50	47.91	47.40	**50.08**
fa (en)	**64.35**	64.04	64.25	63.44	63.92
om (en)	76..27	76.68	76.53	76.2	**77.29**
so (en)	77.63	76.67	77.88	77.88	**78.21**
sw (en)	77.01	77.52	76.84	**77.89**	77.01
yo (en)	68.97	69.21	69.46	70.43	**70.88**
ug (en)	68.73	68.14	68.79	68.69	**69.06**
uz (en)	74.59	73.95	74.46	74.48	**74.75**

Table 7.5: Cross-lingual transfer when the target language has resources (F-score %).

By introducing annotated training data from English, the performance of all the baselines becomes worse than the monolingual baseline. Since the script and sequence structure of English is very different from these low-resource languages, the addition of English to the limited target language training data yields a considerably noisy corpus. However, by forcing the sequence encoder to extract language-agnostic features, our approach still achieves better performance than the monolingual baseline for most languages. All of these experiments demonstrate that our approach is more effective in leveraging annotations from

other languages to improve target language name tagging.

7.9 CROSS-LINGUAL TRANSFER FOR HIGH RESOURCE LANGUAGES

Language	Model	F-score
Dutch	[40]	81.74
	[194]	85.19
	[195]	85.71
	[205]	82.84
	Word-Adv[1]	85.87
	Word-Adv[2]	86.43
	Our Model (Bi-LSTM)	**86.87**
Spanish	[40]	85.75
	[194]	85.77
	[195]	85.02
	[205]	82.95
	Word-Adv[1]	85.92
	Word-Adv[2]	85.84
	Our Model (Bi-LSTM)	**86.41**

Table 7.6: Comparison on cross-lingual transfer for Dutch and Spanish with various baselines: monolingual baseline ([40]), multitask baselines ([194] and [195]), language universal representation baselines ([205], Word-Adv[1], Word-Adv[2]).

We finally investigate the results when both the source and target languages are all high-resource languages. Table 7.6 presents the performance on Dutch and Spanish while using English as the source language. Our approach significantly outperforms all the other approaches even when the size of the annotated training data for the target language is huge. We notice that our approach achieves larger improvement on Dutch than Spanish. The reason may be that, compared with Spanish, Dutch is much closer to English [206]. Both English and Dutch are from the same *West Germanic* branch of the *Indo-European* language family while Spanish is from the *Italic* branch.

7.10 DISCUSSION: IMPACT OF ANNOTATION SIZE FROM SOURCE AND TARGET LANGUAGES

We use Amharic as the target language and Tigrinya as the source language to show the impact of the size of their annotations. Specifically, to explore the impact of the size of target

language annotations, we use 0, 10%, 50%, or 100% annotated training data from Amharic. Similarly, to show the effect of the size of source language annotations, for each experiment, we also gradually add 0, 20%, 50%, or 100% annotated training data from Tigrinya. For all experiments, we use the same dev and test set of Amharic. As Figure 7.3 shows, as we gradually add annotations from the source or target language, the performance can always be improved. When the size of target language annotations is small, such as 400 sentences, we can achieve 5%-30% F-score gain by adding about 4,000 sentences from the source language. When the size of target language annotations is over 2,000 sentences, the improvement is about 2% if we add in about 4,000 sentences from source language annotations.

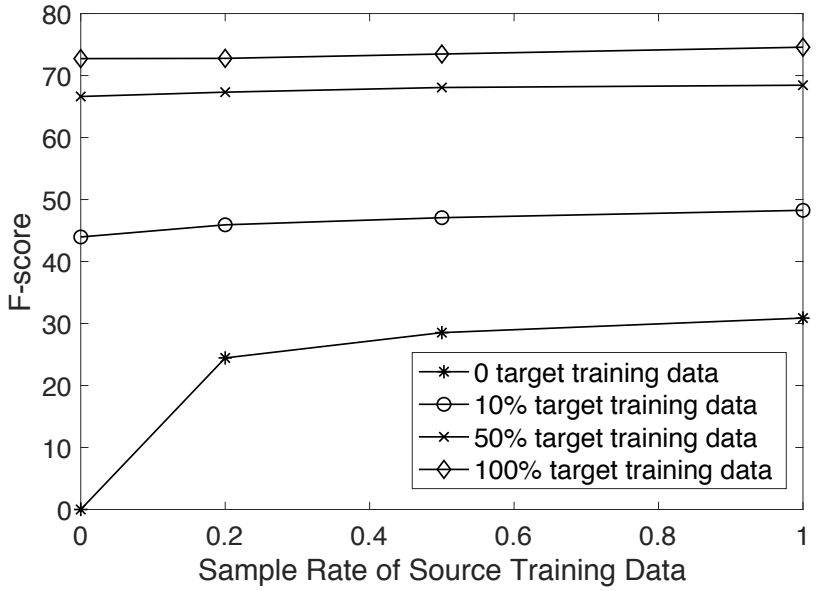

Figure 7.3: The impact of the size of annotations from source and target languages on Amharic name tagging.

7.11 SUMMARY

In this chapter, we integrate a new neural architecture which integrates multi-level adversarial training to learn language-agnostic features to improve low-resource name tagging. With word-level adversarial training, it can automatically project the source language into a shared semantic space with the target language without requiring any comparable data or bilingual gazetteers. Moreover, considering the different underlying sequential structures among various languages, we further design a sentence-level adversarial transfer to encourage the sequence encoder to extract language-agnostic features. The experiments show that

this approach achieves the state-of-the-art on both CoNLL data sets and 10 low-resource languages.

In the future, this framework can be further extended to select the feature-consistent annotations from the source language and add to the target language to further improve cross-lingual low resource name tagging.

CHAPTER 8: CONCLUSIONS, LIMITATIONS, AND FUTURE WORK

Structured data (including entities, events, and their relationships) extracted from natural language text represents the most important information embedded in the data. It can facilitate the understanding of the texts for human so that we can quickly analyze a massive amount of text corpora, discover and reason the key facts that we need. In addition, such structured information can also be aggregated, stored and further managed as background knowledge graph, which is crucial for human to learn and understand the real world. Previous state-of-the-art information extraction approaches heavily rely on human annotated data and can only discover the information for limited predefined types, which render them not to be able to be easily adapted to a new scenario, genre, domain or language, and thus post significant challenges to developing efficient IE algorithms to automatically convert unstructured text into structure knowledge.

8.1 COLD-START UNIVERSAL INFORMATION EXTRACTION: SUMMARY

At the core of this thesis research in Information Extraction (IE) is the desire to endow machines with the ability to automatically extract, assess, and understand text in order to answer the fundamental questions, that is *who did what to whom, when and where*. In particular, the focus of this thesis is on Cold-Start Universal Information Extraction, which establishes a new research direction and creates the next generation of information access in which computers can automatically discover accurate, concise, and trustworthy information embedded in data of any form without requiring any extra human effort. In principal, we develop efficient algorithms and frameworks by exploring the knowledge and connections embedded in symbolic and distributional semantic representations, available ontologies and knowledge bases, as well as other linguistic phenomenons, yielding satisfactory quality and great portability and scalability. The main contributions of this thesis can be summarized in following three aspects:

- We propose a brand new cold-start liberal information extraction paradigm, which moves away the requirement of the high-cost of manual annotations and the narrow-focus of predefined type schemas and bottom-up extracts the structured knowledge from natural language text. Different from the top-down manner of traditional information extraction approaches, The only input to Liberal IE is an arbitrary corpus, without any supervision, restrictions, or prior knowledge of its size, topic, or domain. It can automatically discover the homogeneity of entities and events by leveraging

their rich symbolic and semantic contextual representations, resulting in a corpus-customized and domain-specific type schema. Because of this "cold-start" (or with minimal supervision from existing knowledge bases) fashion, it can be adapted to any domain, genre or language without any extra requirements. The resulting system is being successfully used by various government agencies (e.g., ARL, ARFL, and IARPA) and industrial companies (e.g., Bosch, IBM) on various domains (e.g., military, disaster, bio-medical, power tool). It has also been widely cited and has inspired follow-up research on open-domain information extraction [158, 159, 207, 208, 209, 210, 211], event representation learning [212], event-event relation prediction [213, 214, 215, 216] (Chapter 3).

- We propose a new view of current information extraction problem and accordingly design a new framework to solve it. Traditionally, IE problem, such as entity or event extraction, is viewed as classification, treating each type as a separate class and requiring substantial investment towards annotation or pattern creation. This proposition is impractical when we target at broad types from various genres or domains. Fortunately, my work has shown that pre-defined types can also be encoded by rich contextual or structured representations, through which knowledge elements can be mapped to their appropriate types. Therefore, we frames IE as a grounding problem instead of classification, where knowledge elements are grounded into an extensible and large-scale target ontology with very few available annotations for a few types. Under this new view of IE, I have designed a Zero-Shot learning framework to leverage structured representations of both knowledge elements and types, and embed them into a shared semantic space. To determine if a mention expresses an Attack event, we now ask whether it lies closest to the Attack class in the embedding space. The crucial advantage of this approach is that we only have to train it once because the semantic space and metric is independent of knowledge types and domains, supporting the transfer from old domains (e.g., military action) to new ones (e.g., rescue) with no additional annotation. As a result of these efforts, the extraction capabilities have been extended from dozens of types (e.g., 33 types for event extraction) to more than 1000 types while ensuring high quality. Note that zero-shot IE makes elegant use of all available training data, and is therefore distinct from Open IE, which is type-agnostic and must be mapped to extraction ontologies to be made useful. The system is now being successfully used by ARL. (Chapter 4 and 5)

- We design a new elegant way of transferring available resources, e.g., manually constructed ontology or manually annotated data, from resource-dominant languages (e.g.,

English, Chinese, Spanish) to low resource languages, by constructing a Multi-Lingual Common Semantic Space, which is massively scalable across all languages and is capable of representing words as well as knowledge elements at all levels, from atomic concepts to structured relations and events, in a distributed, language-agnostic continuous semantic space. The key to constructing such a shared space is to generalize from known anchor points and efficiently augment them, since parallel or aligned data between two languages is scarce or unavailable. My solution is called cluster-consistent multi-lingual word embeddings, which constructs a common semantic space by preserving the natural clustering structures of words across multiple languages based on various readily-available linguistic cues such as linguistic properties (e.g., apposition, locative suffixes) derived from knowledge bases that are available for thousands of languages, neighborhood clusters extracted from a monolingual word space, and composed characters which aims to capture similar spelling phenomenon. The resulting embeddings better retain the clustering structures in each language, which is important to multi-lingual IE. This work enables IE to be feasible for thousands of languages without requiring any human effort. By leveraging available resources from English through the common semantic space, we provide coordinated NER (Named Entity Recognition) for hundreds of languages (e.g., Turkish, Amharic, Uyghur) without parallel data and achieve up to 24.5% absolute F-score gain. (Chapter 6 and 7)

Approach	Input	Annotation Requirement	Resource Requirement	Output
Liberal IE [152]	Text Documents	✗	AMR Parser	Knowledge Elements, Type Schema
Zero-Shot IE [217]	Text Documents, Target Ontology	Annotations for a Few Types	AMR Parser	Knowledge Elements for All Types
SS-VQVAE	Text Documents	Annotations for a Few Types	✗	Knowledge Elements, New Types
Common Semantic Space [190]	Word Embeddings	Bilingual Lexicons	Language Universal KBs	Multilingual Common Semantic Space
Cross-Lingual Adversarial Transfer [218]	Text Documents for Target Language	Entity Annotations for Source Language	✗	Entities from Target Language

Table 8.1: Requirements and Outputs of Cold-Start Universal IE Approaches.

To better understand the application scenarios of each framework, we list the input, requirement, as well as output of each approach in Table 8.1. Our Liberal IE approach can be applied where there is no any annotated data or no specified target types, while the

Zero-Shot IE method is applied when there is a target type ontology and a few types are associated with some human annotations. Our SS-VQVAE approach can be further applied when there are some human annotations for a few types, but there is no large-scale target ontology.

8.2 LIMITATIONS

Even with the above innovations that we have developed in this thesis, it's still challenging to automatically construct a high-quality knowledge base without any human effort, especially the quality of the structured knowledge mined with current approaches is still not sufficient. In this chapter, we use following examples shown in Table 8.2 to describe what kind of challenges are still remaining.

ID	Sentence	Category
S1	Three young boys survived and are in **critical** condition after spending 18 hours in the cold.	Polysemy
S2	Today I was **let go** from my job after working there for 4 years.	Polysemy
S3	Still **hurts** me to read this.	Metaphor
S4	Stewart has found the road to fortune wherever she has **traveled**.	Metaphor
S5	When we come back, media speculation run amuck over possible indictments at **sixteen hundred Pennsylvania** and the President's scripted session with troops in Iraq .	Background Knowledge
S6	The **Stockholm Institute** stated that 23 of 25 major armed conflicts in the world in 2000 occurred in impoverished nations.	Commonsense

Table 8.2: Examples about Remaining Challenges.

The first and the fundamental challenge in dealing with language is the **variety** and **ambiguity**. Many language phenomenons, such as Polysemy (e.g., a word or phrase may have multiple meanings), Genericity (e.g., whether an event is specific to a singular occurrence at a particular place and time or is generic to a finite set of such occurrences), Modality (e.g., whether an event is asserted, hypothetical or metaphoric), make it hard for machines to precisely interpret the context. For example, in S1, *critical* is usually used to express adverse or disapprove comments or judgments, however, in this context, it is used to describe a person who is extremely ill and at risk of death, thus it should be identified as an *Injure* event mention. Similarly, *let go* can be used as relinquishing one's grip or dismiss someone, while in S2, it refers to the second meaning thus it should also be identified as an

End Position event mention. Metaphor is omnipresent in our daily language. For example, *hurts* usually refers to an attack event and *traveled* means transport person, however in S3 and S4, both hurt and traveled don't mean the real body hurt or body move. As we can see, to understand the sentence precisely, our approaches need to incorporate more clues and advanced architectures to perform very deep contextual understanding, analysis and reasoning to disambiguate the meaning of the words.

In addition, our algorithms and models are still lack of background knowledge and commonsense to perform precise interpretation of the text and accurate prediction. One of the biggest gaps between human and machines in terms of the understanding of a natural language text lie in the knowledge, such as background knowledge, domain-specific knowledge and commonsense, that they have acquired. This knowledge plays an important role in correctly determining which entry it refers to. For example, in order to determine the type of *sixteen hundred Pennsylvania* in S5, we need to acquire the background knowledge that it refers to the *White House* because it's the physical address of *White House* in Washington D.C.. In S6, *Stockholm Institute* can refer to multiple candidate entries in KB, e.g., *Stockholm International Peace Research Institute* or *Stockholm Institute of Education*. To determine correct target entry, we need to first understand the local context that it's talking about armed conflicts, and according to human common sense, a peace research institute is more likely to talk about armed conflicts than an education institute.

Finally, our IE algorithms and frameworks rely on the multilingual common semantic space as a bridge for knowledge and resource transfer across various languages. However, there are more than 6000 living languages in the real world, and most of the languages have very distinct language phenomenons and properties, which make it extremely hard to efficiently transfer the algorithms, resources and effective patterns across various languages. Taking Amharic and Chinese as an example, they have distinct language scripts and symbols, basic information units, structure dependence such as the SVO (subject-verb-object) structure, and so on, thus these language pairs cannot be aligned well within the common semantic space.

8.3 FUTURE WORK

Considering the limitations of our current approaches that we have discussed, we are going to explore following directions to further improve them.

The ultimate goal of Cold-Start Universal IE is to remove human out of the IE loop. However, even though the algorithms and models we discussed have tremendously reduced the reliance of human effort, they still require some resources, e.g., the liberal IE framework

relies on advanced AMR parsing to detect event mentions and compose the meanings of their contexts, the zero-shot learning approach also requires a target ontology with very high-coverage for the particular scenario. These requisites are not always available for a new domain or language. So the first direction that we want to explore is to develop more efficient algorithms and frameworks to further reduce the requirement of these resources and keep improving the quality of the methods.

As we have discussed, knowledge is crucial for machines to interpret natural language text. For example, the knowledge that *PersonX is very likely to be sentenced and PersonY is likely to be wounded after PersonX attacks PersonY* is helpful for machines to extract the participants of each event and the relationship between two events. However, it's also noticeable that such knowledge is extremely diverse and unlimited, so the second research direction is to explore how to automatically acquire the knowledge, e.g., background knowledge, domain-specific knowledge and commonsense, from large-scale domain specific unlabeled corpora, and integrate this knowledge together with all available knowledge bases to better understand the context and improve the quality of information extraction.

In a lot of cases, the knowledge elements concerning on one particular entity, event or relation are not necessarily included in one single sentence. However, most of the current IE programs are only on sentence level. For example, given the article which is shown in Table 8.3 and is talking about that seven people from Vietnam were convicted and sentenced last week. However, by looking at sentence highlighted in this article, it's not possible to understand where was Hanh sentenced because this information comes very ahead of this sentence. So, the third direction is to extend the current sentence-level information extraction task to document or corpus level, by incorporating coreference resolution, commonsense, and cross-sentence inference.

Seven people convicted last week in Vietnam‘s biggest-ever criminal trial, including two former senior government officials, have requested an appeal of the verdicts, a court official said Tuesday. The trial by a Ho Chi Minh City court was seen as a litmus test of the communist government’s resolve to fight widespread corruption. The “godfather‘’ of organized crime, Truong Van Cam, better known as Nam Cam, was convicted of seven crimes, including murder. He was sentenced to face a firing squad, and his lawyer has said he also plans to appeal. **Hanh, also a former member of the powerful Communist Party Central Committee, was convicted of receiving US$8,500 in bribes from Nam Cam‘s family to secure the crime boss’ early release from labor camp in 1990s. Hanh was sentenced to 10 years in jail.** Chien was convicted of receiving a stereo set worth 27 million dong (US$1,750) from Nam Cam’s family and sentenced to six years in jail.

Table 8.3: Example for Document-Level Information Extraction

CHAPTER 9: REFERENCES

[1] R. Grishman and B. Sundheim, "Message understanding conference-6: A brief history," in *Proc. COLING1996*, 1996.

[2] R. Florian, A. Ittycheriah, H. Jing, and T. Zhang, "Named entity recognition through classifier combination," in *HLT-NAACL*. Association for Computational Linguistics, 2003, pp. 168–171.

[3] S. Miller, J. Guinness, and A. Zamanian, "Name tagging with word clusters and discriminative training," in *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, 2004.

[4] H. Ji and R. Grishman, "Improving name tagging by reference resolution and relation detection," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2005, pp. 411–418.

[5] H. Ji, C. Rudin, and R. Grishman, "Re-ranking algorithms for name tagging," in *Proceedings of the Workshop on Computationally Hard Problems and Joint Inference in Speech and Language Processing*. Association for Computational Linguistics, 2006, pp. 49–56.

[6] H. Ji and R. Grishman, "Knowledge base population: Successful approaches and challenges," in *ACL-HLT*. Association for Computational Linguistics, 2011, pp. 1148–1158.

[7] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "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: Volume 2-Volume 2*. Association for Computational Linguistics, 2009, pp. 1003–1011.

[8] A. Culotta and J. Sorensen, "Dependency tree kernels for relation extraction," in *ACL*. Association for Computational Linguistics, 2004, p. 423.

[9] R. Bunescu and R. J. Mooney, "A shortest path dependency kernel for relation extraction," in *HLT-EMNLP*. Association for Computational Linguistics, 2005, pp. 724–731.

[10] S. Riedel, L. Yao, A. McCallum, and B. M. Marlin, "Relation extraction with matrix factorization and universal schemas," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2013, pp. 74–84.

[11] H. Ji and R. Grishman, "Refining event extraction through cross-document inference," *Proceedings of ACL-08: HLT*, pp. 254–262, 2008.

[12] S. Liao and R. Grishman, "Using document level cross-event inference to improve event extraction," in *Proc. ACL*, 2010.

[13] Q. Li, H. Ji, and L. Huang, "Joint event extraction via structured prediction with global features," in *Proc. ACL*, 2013, pp. 73–82.

[14] Q. Li and H. Ji, "Incremental joint extraction of entity mentions and relations," in *Proceedings ACL*, 2014.

[15] B. Zhang, X. Pan, T. Wang, A. Vaswani, H. Ji, K. Knight, and D. Marcu, "Name tagging for low-resource incident languages based on expectation-driven learning," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 249–259.

[16] B. Zhang, D. Lu, X. Pan, Y. Lin, H. Abudukelimu, H. Ji, and K. Knight, "Embracing non-traditional linguistic resources for low-resource language name tagging," in *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, vol. 1, 2017, pp. 362–372.

[17] X. Pan, B. Zhang, J. May, J. Nothman, K. Knight, and H. Ji, "Cross-lingual name tagging and linking for 282 languages," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2017, pp. 1946–1958.

[18] X. Feng, L. Huang, B. Qin, Y. Lin, H. Ji, and T. Liu, "Multi-level cross-lingual attentive neural architecture for low resource name tagging," *Tsinghua Science and Technology*, vol. 22, no. 6, pp. 633–645, 2017.

[19] M. Fang and T. Cohn, "Model transfer for tagging low-resource languages using a bilingual dictionary," *arXiv preprint arXiv:1705.00424*, 2017.

[20] C.-T. Tsai and D. Roth, "Cross-lingual wikification using multilingual embeddings." in *Proceedings of HLT-NAACL*, 2016.

[21] B. Settles, "Biomedical named entity recognition using conditional random fields and rich feature sets," in *Proceedings of the international joint workshop on natural language processing in biomedicine and its applications*. Association for Computational Linguistics, 2004, pp. 104–107.

[22] R. Leaman and G. Gonzalez, "Banner: an executable survey of advances in biomedical named entity recognition," in *Biocomputing 2008*. World Scientific, 2008, pp. 652–663.

[23] J. Kazama, T. Makino, Y. Ohta, and J. Tsujii, "Tuning support vector machines for biomedical named entity recognition," in *Proceedings of the ACL-02 workshop on Natural language processing in the biomedical domain-Volume 3*. Association for Computational Linguistics, 2002, pp. 1–8.

[24] K.-J. Lee, Y.-S. Hwang, S. Kim, and H.-C. Rim, "Biomedical named entity recognition using two-phase model based on svms," *Journal of Biomedical Informatics*, vol. 37, no. 6, pp. 436–447, 2004.

[25] B. Tang, H. Cao, X. Wang, Q. Chen, and H. Xu, "Evaluating word representation features in biomedical named entity recognition tasks," *BioMed research international*, vol. 2014, 2014.

[26] J. Li, Z. Zhang, X. Li, and H. Chen, "Kernel-based learning for biomedical relation extraction," *Journal of the American Society for Information Science and Technology*, vol. 59, no. 5, pp. 756–769, 2008.

[27] D. Zhou, D. Zhong, and Y. He, "Biomedical relation extraction: from binary to complex," *Computational and mathematical methods in medicine*, vol. 2014, 2014.

[28] J. Björne and T. Salakoski, "Generalizing biomedical event extraction," in *Proceedings of the BioNLP Shared Task 2011 Workshop*. Association for Computational Linguistics, 2011, pp. 183–191.

[29] S. Riedel and A. McCallum, "Fast and robust joint models for biomedical event extraction," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2011, pp. 1–12.

[30] S. Riedel and A. McCallum, "Robust biomedical event extraction with dual decomposition and minimal domain adaptation," in *Proceedings of the BioNLP Shared Task 2011 Workshop*. Association for Computational Linguistics, 2011, pp. 46–50.

[31] D. McClosky, S. Riedel, M. Surdeanu, A. McCallum, and C. D. Manning, "Combining joint models for biomedical event extraction," in *BMC bioinformatics*, vol. 13, no. 11. BioMed Central, 2012, p. S9.

[32] Y. Luan, M. Ostendorf, and H. Hajishirzi, "Scientific information extraction with semi-supervised neural tagging," *arXiv preprint arXiv:1708.06075*, 2017.

[33] Y. Luan, M. Ostendorf, and H. Hajishirzi, "The uwnlp system at semeval-2018 task 7: Neural relation extraction model with selectively incorporated concept embeddings," in *Proceedings of The 12th International Workshop on Semantic Evaluation*, 2018, pp. 788–792.

[34] T. Zhang, H. Li, H. Ji, and S.-F. Chang, "Cross-document event coreference resolution based on cross-media features," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 201–206.

[35] T. Zhang, S. Whitehead, H. Zhang, H. Li, J. Ellis, L. Huang, W. Liu, H. Ji, and S.-F. Chang, "Improving event extraction via multimodal integration," in *Proceedings of the 2017 ACM on Multimedia Conference*. ACM, 2017, pp. 270–278.

[36] E. F. Tjong Kim Sang and F. De Meulder, "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition," in *Proceedings of NAACL-HLT*, 2003.

[37] S. Pradhan, L. Ramshaw, M. Marcus, M. Palmer, R. Weischedel, and N. Xue, "Conll-2011 shared task: Modeling unrestricted coreference in ontonotes," in *Proc. CONLL2011*, 2011.

[38] H. Ji, R. Grishman, H. Dang, K. Griffitt, and J. Ellis, "Overview of the tac 2010 knowledge base population track," in *Proc. TAC2010*, 2010.

[39] H. Ji, R. Grishman, and H. Dang, "An overview of the tac2011 knowledge base population track," in *Proc. TAC2011*, 2011.

[40] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," *arXiv preprint arXiv:1603.01360*, 2016.

[41] X. Ma and E. Hovy, "End-to-end sequence labeling via bi-directional lstm-cnns-crf," *arXiv preprint arXiv:1603.01354*, 2016.

[42] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *arXiv preprint arXiv:1802.05365*, 2018.

[43] L. Sha, F. Qian, B. Chang, and Z. Sui, "Jointly extracting event triggers and arguments by dependency-bridge rnn and tensor-based argument interaction." in *AAAI*, 2018.

[44] O. Etzioni, M. Cafarella, D. Downey, A. Popescu, T. Shaked, S. Soderland, D. Weld, and A. Yates, "Unsupervised named-entity extraction from the web: An experimental study," *Artificial Intelligence*, vol. 165, pp. 91–134, 2005.

[45] M. Banko, M. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni, "Open information extraction for the web," in *Proc. IJCAI2007*, 2007.

[46] M. Banko, O. Etzioni, and T. Center, "The tradeoffs between open and traditional relation extraction," in *Proc. ACL-HLT2008*, 2008.

[47] Y. Shinyama and S. Sekine, "Preemptive information extraction using unrestricted relation discovery," in *Proc. HLT-NAACL2006*, 2006.

[48] S. Sekine, "On-demand information extraction," in *Proc. COLING-ACL2006*, 2006.

[49] M. Collins, "Ranking algorithms for named-entity extraction: Boosting and the voted perceptron," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2002, pp. 489–496.

[50] W. W. Cohen and S. Sarawagi, "Exploiting dictionaries in named entity extraction: combining semi-markov extraction processes and data integration methods," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004, pp. 89–98.

[51] A. Cucchiarelli and P. Velardi, "Unsupervised named entity recognition using syntactic and semantic contextual evidence," *Computational Linguistics*, vol. 27, no. 1, pp. 123–131, 2001.

[52] O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates, "Unsupervised named-entity extraction from the web: An experimental study," *Artificial intelligence*, vol. 165, no. 1, pp. 91–134, 2005.

[53] C. F. Baker and H. Sato, "The framenet data and software," in *Proc. ACL2003*, 2003.

[54] K. Kipper, A. Korhonen, N. Ryant, and M. Palmer, "Extending verbnet with novel verb classes," in *Proc. LREC2006*, 2006.

[55] M. Palmer, D. Gildea, and P. Kingsbury, "The proposition bank: An annotated corpus of semantic roles," *Computational Linguistics*, vol. 31, no. 1, pp. 71–106, 2005.

[56] S. Pradhan, E. Hovy, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel, "Ontonotes: A unified relational semantic representation," *International Journal of Semantic Computing*, 2007.

[57] H. Ji and R. Grishman, "Refining event extraction through cross-document inference," in *ACL*, 2008.

[58] J. Pustejovsky, "The syntax of event structure," *Cognition*, 1991.

[59] M. Artetxe, G. Labaka, and E. Agirre, "Learning bilingual word embeddings with (almost) no bilingual data," in *Proceedings of ACL*, 2017.

[60] M. Artetxe, G. Labaka, and E. Agirre, "A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings," *arXiv preprint arXiv:1805.06297*, 2018.

[61] A. Conneau, G. Lample, M. Ranzato, L. Denoyer, and H. Jégou, "Word translation without parallel data," *arXiv preprint arXiv:1710.04087*, 2017.

[62] M. Atzmueller, P. Kluegl, and F. Puppe, "Rule-based information extraction for structured data acquisition using textmarker." in *LWA*, 2008, pp. 1–7.

[63] A. Mykowiecka, M. Marciniak, and A. Kupść, "Rule-based information extraction from patients' clinical data," *Journal of biomedical informatics*, vol. 42, no. 5, pp. 923–936, 2009.

[64] C. W.-k. Leung, J. Jiang, K. M. A. Chai, H. L. Chieu, and L.-N. Teow, "Unsupervised information extraction with distributional prior knowledge," 2011.

[65] R. Feldman and B. Rosenfeld, "Boosting unsupervised relation extraction by using ner," in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2006, pp. 473–481.

[66] H. L. Guo, L. Zhang, and Z. Su, "Empirical study on the performance stability of named entity recognition model across domains," in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2006, pp. 509–516.

[67] V. Krishnan, S. Das, and S. Chakrabarti, "Enhanced answer type inference from questions using sequential models," in *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, 2005.

[68] A. Yakushiji, Y. Miyao, T. Ohta, Y. Tateisi, and J. Tsujii, "Automatic construction of predicate-argument structure patterns for biomedical information extraction," in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2006, pp. 284–292.

[69] P. Kluegl, M. Toepfer, P.-D. Beck, G. Fette, and F. Puppe, "Uima ruta: Rapid development of rule-based information extraction applications," *Natural Language Engineering*, vol. 22, no. 1, pp. 1–40, 2016.

[70] M. A. Valenzuela-Escárcega, G. Hahn-Powell, and D. Bell, "Odinson: A fast rule-based information extraction framework," in *Proceedings of The 12th Language Resources and Evaluation Conference*, 2020, pp. 2183–2191.

[71] F. Reiss, S. Raghavan, R. Krishnamurthy, H. Zhu, and S. Vaithyanathan, "An algebraic approach to rule-based information extraction," in *2008 IEEE 24th International Conference on Data Engineering*. IEEE, 2008, pp. 933–942.

[72] M. Schmitz, R. Bart, S. Soderland, O. Etzioni et al., "Open language learning for information extraction," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, 2012, pp. 523–534.

[73] L. Yao, A. Haghighi, S. Riedel, and A. McCallum, "Structured relation discovery using generative models," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2011, pp. 1456–1466.

[74] D. P. Putthividhya and J. Hu, "Bootstrapped named entity recognition for product attribute extraction," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2011, pp. 1557–1567.

[75] L. Chiticariu, Y. Li, and F. R. Reiss, "Rule-based information extraction is dead! long live rule-based information extraction systems!" in *EMNLP*, 2013, pp. 827–832.

[76] D. M. Bikel, S. Miller, R. Schwartz, and R. Weischedel, "Nymble: a high-performance learning name-finder," *arXiv preprint cmp-lg/9803003*, 1998.

[77] A. McCallum, D. Freitag, and F. C. Pereira, "Maximum entropy markov models for information extraction and segmentation." in *Icml*, vol. 17, no. 2000, 2000, pp. 591–598.

[78] G. Zhou and J. Su, "Named entity recognition using an hmm-based chunk tagger," in *ACL*. Association for Computational Linguistics, 2002, pp. 473–480.

[79] K. Seymore, A. McCallum, and R. Rosenfeld, "Learning hidden markov model structure for information extraction," in *AAAI-99 workshop on machine learning for information extraction*, 1999, pp. 37–42.

[80] M. Asahara and Y. Matsumoto, "Japanese named entity extraction with redundant morphological analysis," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*. Association for Computational Linguistics, 2003, pp. 8–15.

[81] A. McCallum and W. Li, "Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons," in *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics, 2003, pp. 188–191.

[82] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *ICML*, 2001.

[83] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

[84] I. Segura Bedmar, P. Martínez, and M. Herrero Zazo, "Semeval-2013 task 9: Extraction of drug-drug interactions from biomedical texts (ddiextraction 2013)." Association for Computational Linguistics, 2013.

[85] S. Liu, B. Tang, Q. Chen, and X. Wang, "Effects of semantic features on machine learning-based drug name recognition systems: word embeddings vs. manually constructed dictionaries," *Information*, vol. 6, no. 4, pp. 848–865, 2015.

[86] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 160–167.

[87] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of machine learning research*, vol. 12, no. Aug, pp. 2493–2537, 2011.

[88] A. Graves, N. Jaitly, and A.-r. Mohamed, "Hybrid speech recognition with deep bidirectional lstm," in *2013 IEEE workshop on automatic speech recognition and understanding*. IEEE, 2013, pp. 273–278.

[89] L. Liu, J. Shang, X. Ren, F. F. Xu, H. Gui, J. Peng, and J. Han, "Empower sequence labeling with task-aware neural language model," in *Proceedings of AAAI 2018*, 2018.

[90] M. Sato, H. Shindo, I. Yamada, and Y. Matsumoto, "Segment-level neural conditional random fields for named entity recognition," in *Proceedings of IJCNLP 2017*, 2017, pp. 97–102.

[91] M. Miwa, R. Sætre, Y. Miyao, and J. Tsujii, "A rich feature vector for protein-protein interaction extraction from multiple corpora," in *Proc. EMNLP*, 2009.

[92] B. Liu, L. Qian, H. Wang, and G. Zhou, "Dependency-driven feature-based learning for extracting protein-protein interactions from biomedical text," in *Proc. COLING*, 2010.

[93] Y. Hong, J. Zhang, B. Ma, J. Yao, G. Zhou, and Q. Zhu, "Using cross-entity inference to improve event extraction," in *Proc. ACL*. Association for Computational Linguistics, 2011, pp. 1127–1136.

[94] D. McClosky, M. Surdeanu, and C. D. Manning, "Event extraction as dependency parsing," in *ACL*, 2011, pp. 1626–1635.

[95] R. Sebastian and M. Andrew, "Fast and robust joint models for biomedical event extraction," in *EMNLP*, 2011.

[96] H. Poon and L. Vanderwende, "Joint inference for knowledge extraction from biomedical literature," in *Human language technologies: the 2010 annual conference of the North American chapter of the association for computational linguistics*, 2010, pp. 813–821.

[97] Q. Li, H. Ji, and L. Huang, "Joint event extraction via structured prediction with global features," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2013, pp. 73–82.

[98] Q. Li and H. Ji, "Incremental joint extraction of entity mentions and relations," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2014, pp. 402–412.

[99] Q. Li, H. Ji, Y. Hong, and S. Li, "Constructing information networks using one single model." in *Proc. EMNLP2014*, 2014.

[100] Y. Chen, L. Xu, K. Liu, D. Zeng, and J. Zhao, "Event extraction via dynamic multi-pooling convolutional neural networks," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2015, pp. 167–176.

[101] T. Nguyen, K. Cho, and R. Grishman, "Joint event extraction via recurrent neural networks," in *Proc. NAACL-HLT2016*, 2016.

[102] X. Ren, W. He, M. Qu, **Huang, Lifu**, H. Ji, and J. Han, "Afet: Automatic fine-grained entity typing by hierarchical partial-label embedding," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (**EMNLP 2016**)*, 2016.

[103] L. Liu, X. Ren, Q. Zhu, S. Zhi, H. Gui, H. Ji, and J. Han, "Heterogeneous supervision for relation extraction: A representation learning approach," *arXiv preprint arXiv:1707.00166*, 2017.

[104] L. Hu, L. Zhang, C. Shi, L. Nie, W. Guan, and C. Yang, "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)*, 2019, pp. 3812–3820.

[105] A. Fader, S. Soderland, and O. Etzioni, "Identifying relations for open information extraction," in *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics, 2011, pp. 1535–1545.

[106] F. Wu and D. S. Weld, "Open information extraction using wikipedia," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 118–127.

[107] A. Yates, M. Cafarella, M. Banko, O. Etzioni, M. Broadhead, and S. Soderland, "Textrunner: open information extraction on the web," in *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*. Association for Computational Linguistics, 2007, pp. 25–26.

[108] L. Del Corro and R. Gemulla, "Clausie: clause-based open information extraction," in *Proceedings of the 22nd international conference on World Wide Web*. ACM, 2013, pp. 355–366.

[109] P. Gamallo, M. Garcia, and S. Fernández-Lanza, "Dependency-based open information extraction," in *Proceedings of the Joint Workshop on Unsupervised and Semi-Supervised Learning in NLP*. Association for Computational Linguistics, 2012, pp. 10–18.

[110] O. Etzioni, A. Fader, J. Christensen, S. Soderland, and M. Mausam, "Open information extraction: The second generation." in *Proc. IJCAI2011*, vol. 11, 2011, pp. 3–10.

[111] G. Stanovsky, J. Michael, L. Zettlemoyer, and I. Dagan, "Supervised open information extraction," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 885–895.

[112] L. Cui, F. Wei, and M. Zhou, "Neural open information extraction," *arXiv preprint arXiv:1805.04270*, 2018.

[113] K. Kolluru, S. Aggarwal, V. Rathore, S. Chakrabarti et al., "Imojie: Iterative memory-based joint open information extraction," *arXiv preprint arXiv:2005.08178*, 2020.

[114] A. Roy, Y. Park, T. Lee, and S. Pan, "Supervising unsupervised open information extraction models," 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)*, 2019, pp. 728–737.

[115] J. Beckerman and T. Christakis, "Learning open information extraction of implicit relations from reading comprehension datasets," *arXiv preprint arXiv:1905.07471*, 2019.

[116] M. Kuo, Y. Liang, L. Ji, N. Duan, L. Shou, M. Gong, and P. Chen, "Tag and correct: Question aware open information extraction with two-stage decoding," *arXiv preprint arXiv:2009.07406*, 2020.

[117] S. Zhang, K. Duh, and B. Van Durme, "Mt/ie: Cross-lingual open information extraction with neural sequence-to-sequence models," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 2017, pp. 64–70.

[118] S. Zhang, K. Duh, and B. Van Durme, "Selective decoding for cross-lingual open information extraction," in *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2017, pp. 832–842.

[119] Y. Ro, Y. Lee, and P. Kang, "Multi^ 2oie: Multilingual open information extraction based on multi-head attention with bert," *arXiv preprint arXiv:2009.08128*, 2020.

[120] M. Craven, J. Kumlien et al., "Constructing biological knowledge bases by extracting information from text sources." in *ISMB*, vol. 1999, 1999, pp. 77–86.

[121] A. A. Morgan, L. Hirschman, M. Colosimo, A. S. Yeh, and J. B. Colombe, "Gene name identification and normalization using a model organism database," *Journal of Biomedical Informatics*, vol. 37, no. 6, pp. 396–410, 2004.

[122] R. Snow, D. Jurafsky, and A. Y. Ng, "Learning syntactic patterns for automatic hypernym discovery," in *Advances in neural information processing systems*, 2005, pp. 1297–1304.

[123] T.-V. T. Nguyen and A. Moschitti, "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: short papers-Volume 2*. Association for Computational Linguistics, 2011, pp. 277–282.

[124] S. Takamatsu, I. Sato, and H. Nakagawa, "Reducing wrong labels in distant supervision for relation extraction," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, 2012, pp. 721–729.

[125] M. Surdeanu, J. Tibshirani, R. Nallapati, and C. D. Manning, "Multi-instance multi-label learning for relation extraction," in *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*. Association for Computational Linguistics, 2012, pp. 455–465.

[126] B. Min, R. Grishman, L. Wan, C. Wang, and D. Gondek, "Distant supervision for relation extraction with an incomplete knowledge base," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2013, pp. 777–782.

[127] G. Angeli, J. Tibshirani, J. Wu, and C. D. Manning, "Combining distant and partial supervision for relation extraction," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1556–1567.

[128] D. Zeng, K. Liu, Y. Chen, and J. Zhao, "Distant supervision for relation extraction via piecewise convolutional neural networks," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1753–1762.

[129] C. Quirk and H. Poon, "Distant supervision for relation extraction beyond the sentence boundary," *arXiv preprint arXiv:1609.04873*, 2016.

[130] G. Ji, K. Liu, S. He, J. Zhao et al., "Distant supervision for relation extraction with sentence-level attention and entity descriptions." in *AAAI*, 2017, pp. 3060–3066.

[131] Z. Harris, "Distributional structure," *Word*, vol. 10, no. 23, pp. 146–162, 1954.

[132] K. Kipper, A. Korhonen, N. Ryant, and M. Palmer, "A large-scale classification of english verbs," *Language Resources and Evaluation Journal*, vol. 42, no. 1, pp. 21–40, 2008.

[133] L. Banarescu, C. Bonial, S. Cai, M. Georgescu, K. Griffitt, U. Hermjakob, K. Knight, P. Koehn, M. Palmer, and N. Schneider, "Abstract meaning representation for sembanking," in *Proc. ACL2013 Workshop on Linguistic Annotation and Interoperability with Discourse*, 2013.

[134] D. M. Marie-Catherine, B. M., and C. D. M., "Generating typed dependency parses from phrase structure parses," in *Proceedings LREC*, 2006, pp. 449,454.

[135] C. Wang, N. Xue, and S. Pradhan, "Boosting transition-based amr parsing with refined actions and auxiliary analyzers," in *Proc. ACL2015*, 2015.

[136] D. K. C. D. Manning, "Natural language parsing," in *Advances in Neural Information Processing Systems 15: Proceedings of the 2002 Conference*, vol. 15. MIT Press, 2003, p. 3.

[137] D. Das, D. Chen, A. F. Martins, N. Schneider, and N. A. Smith, "Frame-semantic parsing," *Computational Linguistics*, 2014.

[138] E. Hovy, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel, "Ontonotes: the 90% solution," in *Proc. NAACL2006*, 2006.

[139] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, vol. abs/1301.3781, 2013.

[140] Z. Zhong and H. T. Ng, "It makes sense: A wide-coverage word sense disambiguation system for free text," in *Proceedings of the ACL 2010 System Demonstrations*, 2010, pp. 78–83.

[141] C. Fellbaum, *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[142] W. Yin and H. Schütze, "An exploration of embeddings for generalized phrases," in *Proc. ACL2014 Workshop on Student Research*, 2014.

[143] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning, "Semi-supervised recursive autoencoders for predicting sentiment distributions," in *Proc. EMNLP*, 2011, pp. 151–161.

[144] U. Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[145] Z. Song, A. Bies, S. Strassel, T. Riese, J. Mott, J. Ellis, J. Wright, S. Kulick, N. Ryant, and X. Ma, "From light to rich ere: annotation of entities, relations, and events," in *In Proc. NAACL-HLT2015*, 2015.

[146] Y. Chen, L. Xu, K. Liu, D. Zeng, and J. Zhao, "Event extraction via dynamic multi-pooling convolutional neural networks," in *Proc. ACL2015*, 2015.

[147] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[148] S. Garg, A. Galstyan, U. Hermjakob, and D. Marcu, "Extracting biomolecular interactions using semantic parsing of biomedical text," in *Proc. AAAI*, 2016.

[149] H. Ji, R. Grishman, H. Dang, K. Griffitt, and J. Ellis, "Overview of the tac 2010 knowledge base population track," in *TAC*, 2010.

[150] C. Wang, N. Xue, S. Pradhan, and S. Pradhan, "A transition-based algorithm for amr parsing." in *HLT-NAACL*, 2015.

[151] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. EMNLP2013*, 2013.

[152] L. Huang, T. Cassidy, X. Feng, H. Ji, C. Voss, J. Han, and A. Sil, "Liberal event extraction and event schema induction," in *Proc. ACL2016*, 2016.

[153] X. Feng, L. Huang, D. Tang, B. Qin, H. Ji, and T. Liu, "A language-independent neural network for event detection," in *Proc. ACL2016*, 2016.

[154] Z. Song, A. Bies, S. Strassel, T. Riese, J. Mott, J. Ellis, J. Wright, S. Kulick, N. Ryant, and X. Ma, "From light to rich ere: Annotation of entities, relations, and events," in *Proc. NAACL-HLT2015 Workshop on on EVENTS*, 2015.

[155] M. Palmer, D. Gildea, and N. Xue, "Semantic role labeling," *Synthesis Lectures on Human Language Technologies*, 2010.

[156] N. Chambers, "Event schema induction with a probabilistic entity-driven model." in *EMNLP*, vol. 13, 2013, pp. 1797–1807.

[157] K. Nguyen, X. Tannier, O. Ferret, and R. Besançon, "Generative event schema induction with entity disambiguation," in *Proc. ACL*, 2015.

[158] Q. Yuan, X. Ren, W. He, C. Zhang, X. Geng, L. Huang, H. Ji, C.-Y. Lin, and J. Han, "Open-schema event profiling for massive news corpora," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 587–596.

[159] X. Liu, H.-Y. Huang, and Y. Zhang, "Open domain event extraction using neural latent variable models," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 2860–2871.

[160] L. Huang, T. Cassidy, X. Feng, H. Ji, C. R. Voss, J. Han, and A. Sil, "Liberal event extraction and event schema induction," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2016, pp. 258–268.

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

[162] E. Hovy, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel, "Ontonotes: the 90% solution," in *Proceedings of the human language technology conference of the NAACL, Companion Volume: Short Papers*, 2006, pp. 57–60.

[163] C. F. Baker, C. J. Fillmore, and J. B. Lowe, "The berkeley framenet project," in *Proceedings of the 17th international conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, 1998, pp. 86–90.

[164] A. Gersho and R. M. Gray, *Vector quantization and signal compression*. Springer Science & Business Media, 2012, vol. 159.

[165] A. van den Oord, O. Vinyals et al., "Neural discrete representation learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 6306–6315.

[166] B. Yang and T. Mitchell, "Joint extraction of events and entities within a document context," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 289–299.

[167] X. Liu, Z. Luo, and H.-Y. Huang, "Jointly multiple events extraction via attention-based graph information aggregation," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 1247–1256.

[168] T. M. Nguyen and T. H. Nguyen, "One for all: Neural joint modeling of entities and events," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 6851–6858.

[169] H. Yan, X. Jin, X. Meng, J. Guo, and X. Cheng, "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)*, 2019, pp. 5770–5774.

[170] X. Wang, X. Han, Z. Liu, M. Sun, and P. Li, "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)*, 2019, pp. 998–1008.

[171] Y. Lin, H. Ji, F. Huang, and L. Wu, "A joint neural model for information extraction with global features," 2020.

[172] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl et al., "Constrained k-means clustering with background knowledge," in *Icml*, vol. 1, 2001, pp. 577–584.

[173] S. Chandar, M. M. Khapra, H. Larochelle, and B. Ravindran, "Correlational neural networks," *Neural computation*, 2016.

[174] J. Rajendran, M. M. Khapra, S. Chandar, and B. Ravindran, "Bridge correlational neural networks for multilingual multimodal representation learning," *arXiv preprint arXiv:1510.03519*, 2015.

[175] Y. Tsvetkov, M. Faruqui, W. Ling, G. Lample, and C. Dyer, "Evaluation of word vector representations by subspace alignment," in *Proceedings of EMNLP*, 2015.

[176] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, "Character-aware neural language models." in *Proceedings of AAAI*, 2016.

[177] W. Ammar, G. Mulcaire, Y. Tsvetkov, G. Lample, C. Dyer, and N. A. Smith, "Massively multilingual word embeddings," *arXiv preprint arXiv:1602.01925*, 2016.

[178] L. Duong, H. Kanayama, T. Ma, S. Bird, and T. Cohn, "Multilingual training of crosslingual word embeddings," in *Proceedings of EMNLP*, 2017.

[179] M. Faruqui and C. Dyer, "Improving vector space word representations using multilingual correlation," in *Proceedings of EMNLP*, 2014.

[180] T. Luong, H. Pham, and C. D. Manning, "Bilingual word representations with monolingual quality in mind." in *Proceedings of HLT-NAACL*, 2015.

[181] G. A. Miller, C. Leacock, R. Tengi, and R. T. Bunker, "A semantic concordance," in *Proceedings of HLT*, 1993.

[182] Z. Huang, W. Xu, and K. Yu, "Bidirectional lstm-crf models for sequence tagging," *arXiv preprint arXiv:1508.01991*, 2015.

[183] D. Yarowsky, G. Ngai, and R. Wicentowski, "Inducing multilingual text analysis tools via robust projection across aligned corpora," in *Proceedings of the first international conference on Human language technology research*. Association for Computational Linguistics, 2001, pp. 1–8.

[184] M. Wang, W. Che, and C. D. Manning, "Joint word alignment and bilingual named entity recognition using dual decomposition," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2013, pp. 1073–1082.

[185] M. Fang and T. Cohn, "Learning when to trust distant supervision: An application to low-resource pos tagging using cross-lingual projection," *arXiv preprint arXiv:1607.01133*, 2016.

[186] M. Ehrmann, M. Turchi, and R. Steinberger, "Building a multilingual named entity-annotated corpus using annotation projection," in *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, 2011, pp. 118–124.

[187] J. Ni, G. Dinu, and R. Florian, "Weakly supervised cross-lingual named entity recognition via effective annotation and representation projection," *arXiv preprint arXiv:1707.02483*, 2017.

[188] A. Zirikly and M. Hagiwara, "Cross-lingual transfer of named entity recognizers without parallel corpora," in *Proceedings of ACL 2015*, 2015.

[189] D. Wang, N. Peng, and K. Duh, "A multi-task learning approach to adapting bilingual word embeddings for cross-lingual named entity recognition," in *Proceedings of IJCNLP 2017*, 2017.

[190] L. Huang, K. Cho, B. Zhang, H. Ji, and K. Knight, "Multi-lingual common semantic space construction via cluster-consistent word embedding," *arXiv preprint arXiv:1804.07875*, 2018.

[191] S. Kim, K. Toutanova, and H. Yu, "Multilingual named entity recognition using parallel data and metadata from wikipedia," in *Proceedings of ACL 2012*, 2012, pp. 694–702.

[192] J. Nothman, N. Ringland, W. Radford, T. Murphy, and J. R. Curran, "Learning multilingual named entity recognition from wikipedia," *Artificial Intelligence*, pp. 151–175, 2013.

[193] Z. Yang, R. Salakhutdinov, and W. Cohen, "Multi-task cross-lingual sequence tagging from scratch," *arXiv preprint arXiv:1603.06270*, 2016.

[194] Z. Yang, R. Salakhutdinov, and W. W. Cohen, "Transfer learning for sequence tagging with hierarchical recurrent networks," *arXiv preprint arXiv:1703.06345*, 2017.

[195] Y. Lin, S. Yang, V. Stoyanov, and H. Ji, "A multi-lingual multi-task architecture for low-resource sequence labeling," in *Proceedings of ACL 2018*, vol. 1, 2018, pp. 799–809.

[196] P. Koehn, "Europarl: A parallel corpus for statistical machine translation," in *MT summit*, vol. 5, 2005, pp. 79–86.

[197] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of NIPS 2012*, 2012, pp. 1097–1105.

[198] M. Zhang, Y. Liu, H. Luan, and M. Sun, "Adversarial training for unsupervised bilingual lexicon induction," in *Proceedings of ACL*, 2017.

[199] X. Chen and C. Cardie, "Unsupervised multilingual word embeddings," in *Proceedings of EMNLP 2018*, 2018.

[200] C. Xing, D. Wang, C. Liu, and Y. Lin, "Normalized word embedding and orthogonal transform for bilingual word translation." in *Proceedings of HLT-NAACL*, 2015.

[201] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proceedings of NIPS 2014*, 2014.

[202] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT 2010*, 2010.

[203] H. Ji, J. Nothman, B. Hachey, and R. Florian, "Overview of tac-kbp2015 tri-lingual entity discovery and linking," in *Proceedings of TAC 2015*, 2015.

[204] E. F. Tjong Kim Sang, "Introduction to the conll-2002 shared task: Language-independent named entity recognition," in *Proceedings of the 6th Conference on Natural Language Learning - Volume 20*, ser. COLING-02. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002. [Online]. Available: https://doi.org/10.3115/1118853.1118877 pp. 1–4.

[205] D. Gillick, C. Brunk, O. Vinyals, and A. Subramanya, "Multilingual language processing from bytes," in *Proceedings of NAACL-HLT 2016*, 2016.

[206] A. Cutler and D. Pasveer, "Explaining cross-linguistic differences in effects of lexical stress on spoken-word recognition," in *3rd International Conference on Speech Prosody*. TUD press, 2006.

[207] J. Araki and T. Mitamura, "Open-domain event detection using distant supervision," in *Proceedings of the 27th International Conference on Computational Linguistics (**COLING 2018**)*, 2018.

[208] Z. Zhang, Y. Wu, and Z. Wang, "A survey of open domain event extraction."

[209] Y. Zeng, Y. Feng, R. Ma, Z. Wang, R. Yan, C. Shi, and D. Zhao, "Scale up event extraction learning via automatic training data generation," in *Thirty-Second AAAI Conference on Artificial Intelligence (**AAAI 2018**)*, 2018.

[210] Z. Yuan and D. Downey, "Otyper: A neural architecture for open named entity typing," in *Thirty-Second AAAI Conference on Artificial Intelligence (**AAAI 2018**)*, 2018.

[211] B. Shi and T. Weninger, "Open-world knowledge graph completion," in *Thirty-Second AAAI Conference on Artificial Intelligence (**AAAI 2018**)*, 2018.

[212] N. Weber, N. Balasubramanian, and N. Chambers, "Event representations with tensor-based compositions," in *Thirty-Second AAAI Conference on Artificial Intelligence (**AAAI 2018**)*, 2018.

[213] A. Badgett and R. Huang, "Extracting subevents via an effective two-phase approach," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (**EMNLP 2016**)*, 2016.

[214] T. Mitamura, Z. Liu, and E. H. Hovy, "Events detection, coreference and sequencing: What's next? overview of the tac kbp 2017 event track." in *TAC*, 2017.

[215] H. Peng, "Understanding stories via event sequence modeling," Ph.D. dissertation, University of Illinois at Urbana-Champaign, 2018.

[216] H. Peng, Y. Song, and D. Roth, "Event detection and co-reference with minimal supervision," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 392–402.

[217] L. Huang, H. Ji, K. Cho, and C. R. Voss, "Zero-shot transfer learning for event extraction," *arXiv preprint arXiv:1707.01066*, 2017.

[218] **Huang, Lifu**, H. Ji, and J. May, "Cross-lingual multi-level adversarial transfer to enhance low-resource name tagging," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (**NAACL 2019**)*, 2019.