

JOINT INFORMATION EXTRACTION

By

Qi Li

A Dissertation Submitted to the Graduate
Faculty of Rensselaer Polytechnic Institute

in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Major Subject: COMPUTER SCIENCE

Examining Committee:

Heng Ji, Dissertation Adviser

James A. Hendler, Member

Peter Fox, Member

Dan Roth, Member

Daniel M. Bikel, Member

Rensselaer Polytechnic Institute
Troy, New York

April 2015
(For Graduation May 2015)

© Copyright 2015
by
Qi Li
All Rights Reserved

CONTENTS

LIST OF TABLES	vi
LIST OF FIGURES	viii
LIST OF ABBREVIATIONS	ix
ACKNOWLEDGMENT	xi
ABSTRACT	xiii
1. Introduction	1
1.1 Introduction of Information Extraction	1
1.1.1 Entity Mention	4
1.1.2 Relation	4
1.1.3 Event	5
1.1.4 Graphical Example	6
1.2 Joint Information Extraction	7
1.3 Related Publications	10
1.4 Thesis Structure	10
2. Related Work	11
2.1 Entity Mention Extraction, Relation Extraction, and Event Extraction	11
2.2 Utilizing Dependencies Across Languages	12
2.3 Joint Event Extraction	13
2.4 Joint Extraction of Entity Mentions and Relations	13
2.5 Other Joint Modeling Methods	14
3. Joint Information Extraction Framework with Structured Prediction and Inexact Search	16
3.1 Problem Definition and Motivation	16
3.2 Baseline Systems	19
3.2.1 Entity Mention Classifier	20
3.2.2 Relation Classifier	21
3.2.3 Event Mention Pipeline	21
3.3 Joint Extraction Framework	21

3.3.1	Structured Perceptron with Beam Search	23
3.3.2	Decoding Algorithms	26
3.3.3	Global Feature vs. Local Feature	29
3.3.4	Entity Type Constraints	30
3.4	Token-based Decoding	31
3.4.1	Decoding Example	33
3.5	Segment-based Decoding	33
3.5.1	Decoding Example	37
3.6	Features	38
3.6.1	Local Features	39
3.6.1.1	Trigger Features	39
3.6.1.2	Argument Features	41
3.6.1.3	Entity Mention Features	41
3.6.2	Global Features	43
3.6.2.1	Global Trigger Features	43
3.6.2.2	Global Argument Features	44
3.6.2.3	Global Entity Mention Features	45
3.6.2.4	Global Relation Features	47
3.6.2.5	Joint Relation-Event Feature	48
3.7	Experiments	49
3.7.1	Evaluate Setup	49
3.7.2	Results of Token-based Decoding	51
3.7.3	Results of End-to-End Relation Extraction	54
3.7.4	Results of Complete Joint Model	56
3.7.5	Real Example	59
3.8	Analysis of Human Performance	60
3.9	Analysis of Remaining Challenges	63
3.10	Discussion	66
4.	Joint Inference for Cross-document Information Extraction	67
4.1	Cross-document Information Extraction	67
4.2	Experiments	71
4.2.1	Overall Performance	71
4.2.2	Impact of Different Types of Constraints	72
4.3	Discussion	75

5. Joint Bilingual Name Tagging	76
5.1 Baseline Approach	76
5.2 Joint Bilingual Name Tagger	77
5.2.1 Linear-chain CRF with Cross-lingual Features	77
5.2.2 Bilingual Conditional Random Fields	79
5.3 Experiments	82
5.3.1 Evaluation Setup	82
5.3.2 Overall Performance	83
5.3.3 Learning Curves	84
5.4 Extrinsic Evaluation on Name-aware Machine Translation	85
5.4.1 Evaluation Setup	86
5.4.2 Overall Performance	87
5.4.3 Improving Word Alignment	87
5.4.4 Analysis	89
5.5 Discussion	90
6. Conclusions and Future Directions	91
REFERENCES	94

LIST OF TABLES

1.1	Examples of entity types.	4
1.2	Examples of relation types.	5
1.3	Example of event mention.	6
3.1	Features in event trigger detection.	22
3.2	Features in event argument labeling.	22
3.3	Comparison between token-based decoding and segment-based decoding.	28
3.4	Examples of entity-type constraints in ACE'05 corpus.	30
3.5	Global trigger features.	43
3.6	Top five event subtypes that co-occur with Attack event.	44
3.7	Global argument features.	44
3.8	Frequent overlapping relation and event types.	48
3.9	Statistics about data set.	51
3.10	Comparison of training time and accuracy on dev set.	52
3.11	Overall performance with gold-standard argument candidates (entity mention, ACE value, and timex).	53
3.12	Overall performance from 5-fold cross-validation.	53
3.13	Comparison between the performance (%) of standard-update and early-update with global features.	54
3.14	Overall performance on ACE'05 corpus.	55
3.15	5-fold cross-validation on ACE'04 corpus.	56
3.16	Overall performance (%) on test set.	57
3.17	Top features about event triggers.	58
4.1	Examples of incompatible constraints.	69
4.2	Numbers of unique relation and event predicates.	72
4.3	Impact of different constraints on Member_of and Family	73

5.1	Monolingual features in baseline systems.	78
5.2	The number of names in the bilingual data set.	83
5.3	Performance (%) on bilingual data set.	84
5.4	Translation performance (%).	88
5.5	Impact of joint bilingual name tagging on word alignment.	89

LIST OF FIGURES

1.1	Overview of the main objectives of this thesis.	8
3.1	Examples of entity mention and relation extraction from ACE'04 corpus.	17
3.2	Information network representation.	18
3.3	Overview of pipelined approach.	20
3.4	Overview of joint extraction framework.	23
3.5	Example of early update in beam search.	25
3.6	Perceptron learning algorithm with beam-search and early-update.	26
3.7	Example of event mentions.	31
3.8	Example notation with $s = 3, m = 2$	31
3.9	Token-based decoding algorithm.	32
3.10	Example of decoding steps.	34
3.11	Segment-based decoding algorithm.	35
3.12	Example of decoding steps.	37
3.13	Distribution of triggers and their frames.	40
3.14	Segment-based local entity mention features.	42
3.15	Illustration of the global features in Table 3.7.	46
3.16	Examples of global relation features.	48
3.17	Distribution of event trigger w.r.t. POS tags.	49
3.18	Training curves on development set.	52
3.19	Percentage of “invalid updates” in standard perceptron.	53
3.20	Learning curves on development set.	55
3.21	Two competing hypotheses for “ <i>a marcher from Florida</i> ” during joint extraction with global features.	59
3.22	Distribution of different types of challenges.	63

4.1	Dependency constraints over entities and their links.	68
4.2	Browsing cost comparison of <code>Member_Of</code> and <code>Family</code> relations.	72
4.3	Removal curves w.r.t parameter θ	73
5.1	Example of parallel sentence pair.	77
5.2	Graphical representation of bilingual CRF model.	81
5.3	Performance on different sizes of training data.	84
5.4	Word alignment gains according to the percentage of name words in each sentence.	88

LIST OF ABBREVIATIONS

IE	Information Extraction.
ACE	Automatic Content Extraction.
NER	Named Entity Recognition, or Name Tagging.
SVM	Support Vector Machine.
CRF	Conditional Random Field.
MaxEnt	Maximum Entropy (Classifier).
ILP	Integer Linear Programming.
MIRA	Margin Infused Relaxed Algorithm.
PMI	Point-wise Mutual Information.
POS	Part of Speech.

To my grandparents.

ACKNOWLEDGMENT

I am indebted to many individuals who have advised, supported, and influenced me in my journey of graduate study. I cannot express my gratitude enough for their contributions to my progress.

First and foremost, I would like to thank my advisor, Heng Ji, for teaching me to be an independent researcher. I will never forget the moments that she patiently helped me improving my writing and presentation skills, and inspired my research ideas. She can always guide me the right directions and point me the right resources to refer to. Most importantly, her passion for scientific research has deeply influenced me since my first day in our group. There is no doubt that without her advising, support, and encouragement, I could not have accomplished this thesis.

I am greatly honored to have the wonderful researchers and mentors in my doctoral committee: James Hendler, Peter Fox, Dan Roth and Dan Bikel. My thesis work is largely motivated and influenced by their research. I would like to thank all of them for their insightful comments and suggestions for my thesis.

I feel very lucky to be a member of blender lab. I will never forget those days that I worked with my colleagues to solve challenging problems in many research projects. I would like to give special thanks to Zheng Chen, Haibo Li, Xiang Li, Hao Li, Taylor Cassidy, and Javier Artiles for their productive and supportive collaborations with me. I also owe gratitude to Liang Huang, Sujian Li for their advising and support for my thesis work. I learned tremendous knowledge and skills from all of them, which then shaped my path of graduate study. I was also lucky to work with Faisal Farooq, Sugato Bagchi, Siddharth Patwardhan, Gokhan Tur, and Dilek Hakkani-Tür during my summer internships. My progress has benefited from them tremendously.

Finally, I thank my parents and grandparents for their support for me to pursue PhD degree. Finally, I want to thank my wife, Yongjie Cai, for her selfless love, which encouraged me to overcome every setback during my graduate life.

ABSTRACT

Information extraction (IE) is a challenging and essential task in the area of natural language processing (NLP), and can be applied to a broad range of applications such as question answering, conversational language understanding, machine translation and many more. It aims to automatically identify important entity mentions and their interactions such as relations and events from unstructured documents. In the past decade, researchers have made significant progress in this area. Although many IE approaches employ a pipeline of many independent components, various dependencies in IE from *multiple components*, *multiple documents*, and *multiple languages* are pervasive. The ignorance of those dependencies in traditional approaches leads to inferior performance because of the fact that the local classifications do not talk to each other to produce coherent results, and more importantly, they are incapable of performing global inference. Therefore it is critical to devise *cross-component*, *cross-document*, and *cross-language* joint modeling methods to further improve the performance of IE.

Taking entity mention extraction, relation extraction and event extraction as points of view, the main part of this thesis presents a novel sentence-level joint IE framework based on structured prediction and inexact search. In this new framework, the three types of IE components can be simultaneously extracted to alleviate error propagation problem. And we can make use of various global features to produce more accurate and coherent results. Experimental results on the ACE corpora show that our joint model achieves state-of-the-art performance on each stage of the extraction. We further go beyond sentence level and make improvement in cross-document setting. We use an integer-linear-programming (ILP) formulation to conduct cross-document inference so that many spurious results can be effectively filtered out based on the inter-dependencies over the facts from different places. Finally, to investigate the cross-lingual dependencies, we present a CRF-based joint bilingual name tagger for parallel corpora, then demonstrate the application of this method to enhance name-aware machine translation.

CHAPTER 1

Introduction

Information extraction (IE) is an important task in the field of Natural Language Processing (NLP), and has been applied to various applications such as question answering, information retrieval, conversational language understanding, machine translation and many more. The goal of IE is to extract information structures of entity mentions and their interactions such as relations and events from unstructured documents. The task is often artificially broken down into several subtasks, and various types of facts are extracted in isolation. Errors in upstream components are propagated to the downstream classifiers, often resulting in compounding errors. However, various entity mentions and their interactions in the information structure are inter-dependent. Also the output structures should comply with multiple soft and hard constraints. In this thesis we study the topic of joint information extraction to bridge the gap among multiple local predictions in traditional approaches, and produce more accurate and coherent IE results. In this chapter, we begin by describing the subtasks in information extraction that we addressed on in this thesis (Section 1.1), and then briefly introduce the main concepts of joint information extraction (Section 1.2), and finally overview the structure of this thesis (Section 1.4)

1.1 Introduction of Information Extraction

In this section, we explain to the reader the essential background knowledge of this thesis. The IE tasks that we are addressing are those of the Automatic Content Extraction (ACE) program¹.

IE is the task of identifying and classifying entities that are mentioned in natural language documents, and the predicates and attributes that they are associated with, such as relations and events. Some common sub-tasks include entity mention extraction, relation extraction, and event extraction, slot filling, entity linking, co-reference resolution etc. In this thesis, we are focusing on the three most

¹<http://www.nist.gov/speech/tests/ace> (Date Last Accessed, March, 10, 2015)

fundamental and important tasks: entity mention extraction, relation extraction and event extraction. With the help of these techniques, we can ask computers to automatically process massive amount of natural language documents such as news articles and web blogs, and render the important facts to potential users. For instance, let us consider the following excerpt of *Marissa Mayer*'s Wikipedia page:

Marissa Ann Mayer is the current president and CEO of Yahoo!.

...

Mayer was the Vice President of Google Product Search until the end of 2010, when she was moved by then-CEO Eric Schmidt.

On July 16, 2012, Mayer was appointed President and CEO of Yahoo!.

...

Mayer married lawyer and investor Zachary Bogue on December 12.

...

and Mayer gave birth to a baby boy on September 30, 2012.

While human being can fully understand those sentences and capture the important information from them, it is challenging for computers to distill the semantics. By developing IE algorithms, we can enable computers to extract the following structured information automatically:

Entity Mention	Type
"Marissa Ann Mayer"	Person
"Eric Schmidt"	Person
"Google"	Organization
"baby"	Person
"Yahoo!"	Organization
"Zachary Bogue"	Person

This information can then be processed by downstream applications such as question answering, document summarization, and information retrieval. The table above only contains persons and organizations. It is more useful to acquire the static relations and dynamic events that they participate in. For example, if we know

“Mayer” and “Zachary Bogue” were participants in a *Marriage* event, and the relation between “Mayer” and the company “Yahoo!” is *CEO*, then in a question answering system we will be able to answer questions such as “(who is the) spouse of Marissa Mayer”, or “(who is the) CEO of Yahoo!”. To this end, the tasks of relation extraction and event extraction aim to discover the following information:

“Marissa A. Mayer” (Employee)	Employment (president, CEO)	“Yahoo!” (Employer)
“Eric Schmidt” (Employee)	Employment (CEO)	“Google” (Employer)
“Mayer” (Person)	Start-Position (appointed)	“Yahoo!” (Entity)
“she” (Person)	End-Position (moved)	“Google” (Entity)
“Mayer” (Argument-1)	Marry (married)	“Zachary Bogue” (Argument-2)
“Mayer” (Argument-1)	Birth (gave birth)	“baby” (Argument-2)

With all of the information above, we obtain a knowledge graph about the entities that are mentioned in the document. In later chapters, we refer to it as *information networks*. It worth noting that in addition to the above information, it is also important to determine that “she” and “Marissa Ann Mayer” refer to the same person. This can be handled by entity co-reference resolution [1], which is another challenging and extensively studied task in IE and beyond the scope of this thesis.

IE has been a popular research area since 1990s, when the first Message Understanding Conference (MUC-1) was introduced. Standard human-annotated corpora and evaluation metrics was developed along with the evolutions from MUC-1 to MUC-7 [2]. Automatic Content Extraction (ACE)² then made significant progress on covering a broad range of entities, relations, and events from multiple genres (such as news articles, broadcast, and web forums), several topics (such as politics, sports, finance etc.), and multiple languages including English, Chinese, and Arabic. Knowledge Base Population (KBP)³ further moved from single-document IE tasks to cross-document tasks. For example, in KBP Slot Filling, given an entity mention as a query, a system is required to look for pre-defined attributes (such as residence, spouse etc.) for the query from a large collection of unlabeled documents. However, many systems rely on entity mention extraction, relation extraction and event extraction [3].

²<http://www.itl.nist.gov/iad/mig/tests/ace/> (Date Last Accessed, March, 10, 2015)

³<http://www.nist.gov/tac/2014/KBP/index.html> (Date Last Accessed, March, 10, 2015)

Table 1.1: Examples of entity types.

Entity Type	Example
Person (PER)	<u>Saddam Hussein</u> 's regime has fallen.
Organization (ORG)	Downer told the <u>Australian Broadcasting Corp.</u>
Geographical Entities (GPE)	<u>U.S.</u> marines entered southeastern Baghdad.
Location (LOC)	Along the <u>riverfront</u> there.
Facility (FAC)	sent her to the <u>hospital</u> .
Weapon (WEA)	two <u>bullets</u> hit the windshield.
Vehicle (VEH)	an <u>airline</u> crashed near New York.

1.1.1 Entity Mention

An *entity mention* is a reference to an object or a set of objects in the world. It may be a reference by its name, a common noun or noun phrase, or a pronoun. There are three different major mention types: 1) *Named mention* (NAM): proper names of entities such as “Steve Jobs”, “Apple Inc.” and “California”; 2) *Nominal mention* (NOM): such as “the executive”, “the company” and “some states”; and 3) *Pronoun mention* (PRO): such as “he”, “they”, and “it”. The task of identification and classification of named entities is commonly known as *named entity recognition* (NER) or *name tagging*. In this thesis, those two names are interchangeable unless otherwise noted. [4] gives a comprehensive literature survey about this subtask.

Different from the notion of mention type, *entity type* describes the type of the entity that an entity mention refers to. ACE defined 7 main entity types including: Person (PER), Organization (ORG), Geographical Entities (GPE), Location (LOC), Facility (FAC), Weapon (WEA) and Vehicle (VEH). Each of them has a certain number of fine-grained subtypes. For example, Organization contains 9 subtypes including Sports Organization, Government Organization, Religious Organization, Media Organization. In this thesis, we only focus on the main types. Table 1.1 shows some examples of the 7 main types.

1.1.2 Relation

A relation⁴ is a directed semantic relation of the targeted types between a pair of entity mentions that appear in the same sentence. ACE’04 defined 7 main re-

⁴Throughout this thesis we refer to relation mention as relation since we do not consider relation mention co-reference. Similarly, we refer to event mention as event.

Table 1.2: Examples of relation types.

Relation Type	Example
Physical	<u>Stanley McChrystal</u> _(PER) said at the <u>Pentagon</u> _(GPE) Tuesday.
Part-whole	the key regime <u>centers</u> _(LOC) of power in <u>Baghdad</u> _(GPE) .
Personal-Social	<u>Nathan</u> _(PER) divorced wallpaper salesman <u>Bruce Nathan</u> _(PER)
Agent-Artifact	<u>North Korea</u> _(ORG) 's <u>weapons</u> _(WEP)
ORG-Affiliation	The <u>tire maker</u> _(ORG) still employs <u>1,400</u> _(PER)
Gen-Affiliation	<u>Israeli</u> _(GPE) forensics <u>experts</u> _(PER)

lation types: Physical (PHYS), Person-Social (PER-SOC), Employment-Organization (EMP-ORG), Agent-Artifact (ART), PER/ORG Affiliation (Other-AFF), GPE-Affiliation (GPE-AFF) and Discourse (DISC). ACE'05 kept PER-SOC, ART and GPE-AFF, split PHYS into PHYS and a new relation Part-Whole, removed DISC, and merged EMP-ORG and Other-AFF into ORG-Affiliation. Table 1.2 demonstrates some examples of relations, where underscore indicates the two arguments for each relation.

1.1.3 Event

An event is a specific occurrence of event with several participants. ACE'05 defined 8 event types and 33 subtypes such as *Attack*, *End-Position* etc. We introduce the terminology of the ACE event extraction as follows:

- Event mention: an occurrence of an event in the text with a particular type, a trigger (a.k.a. anchor), and a set of arguments.
- Event trigger: the word that most clearly expresses the event mention.
- Event argument: an entity mention, temporal expression or value (e.g. *Job-Title*) serves as a participant or attribute with a specific role for an event mention. Each type of event has a set of predefined argument roles. For example, *Transport* event has *Agent*, *Artifact*, *Vehicle*, *Origin*, *Destination* and *Time*. Some argument roles are shared by multiple types of event. For example, *Place* and *Time* are possible arguments to all types of events.

According to the ACE definition, the arguments and the trigger of each event mention should appear in the same sentence. Table 1.3 shows an example of two event mentions in the sentence “*Greece began evacuating its embassy in Baghdad*”. We

Table 1.3: Example of event mention.

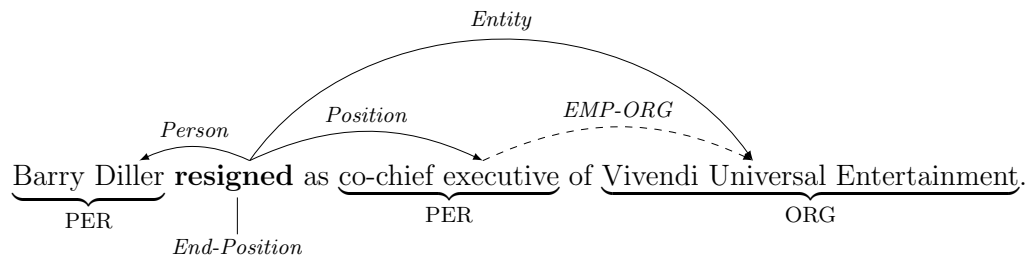
Type	Transport		
Trigger	“evacuate”		
Arguments	Argument Role	Mention Head	Entity Type
	Artifact	“embassy”	ORG
	Origin	“Baghdad”	GPE
	Agent	“Greece”	GPE

refer the reader to the official ACE annotation guideline for a complete list of event types and argument roles [5]. To summarize, there are two main differences between events and relations:

1. Relations are static or long-lasting predicates between entity mentions, while events capture dynamic activities that entity mentions participate in.
2. In each relation, there exist exactly two arguments. On the contrary, each event can have arbitrary numbers of arguments. For instance, although **Transport** event has six possible types of arguments, in Table 1.3, we only tag those occur in the same sentence.

1.1.4 Graphical Example

In the previous examples, entity mentions, relations and events are all described by tables with disconnected entries as in most prior work (such as [6] and [7]). It is more reasonable and convenient to represent them in a graph, especially when multiple types of information occur in the same sentence. For instance, for the sentence “*Barry Diller resigned as co-chief executive of Vivendi Universal Entertainment.*” An information extractor should produce the entity mention, relation, event mention structure as depicted in the following figure:



where solid lines represent event trigger “*resigned*” and its arguments, and dashed represents the relation between Person mention “*co-chief executive*” and Organization mention “*Vivendi Universal Entertainment*”. In later chapters, we will formulate the IE annotation to be “*information networks*”, which is consistent with this representation.

1.2 Joint Information Extraction

In the realm of IE, dependencies and constraints across *multiple components*, *multiple documents*, and *multiple languages* are pervasive. In this thesis, we improve the performance of IE by utilizing those dependencies in joint extraction frameworks. Figure 1.1 gives an overview of the main objectives in this thesis.

Most state-of-the-art IE approaches [6–12] used sequential pipelines as building blocks, and break down the whole task into multiple separate subtasks. For instance, to build an event extraction or relation extraction system, they first need to obtain entity mention information from a separate classifier or manual annotation. Additionally, the extraction of event triggers and argument links are regarded as two isolated subtasks. As a result, a full pipeline of IE is comprised of many isolated local classifiers. We argue that such a pipelined approach has the following limitations: First, it prohibits the interactions among components. Errors in upstream components are propagated to the downstream classifiers, often resulting in compounding errors. The downstream components, however, cannot impact earlier decisions with feedback. Second, it over-simplifies the whole task as a set of isolated local classification problems without taking into account global dependencies. When skilled human readers distill information from documents, they do not predict different types of facts step by step in a “bottom-up” fashion; instead, they tend to take a “top-down” approach - attempting to comprehend content and predict high-level information through the lens of prior knowledge about the general subject before making sense of the details [13, 14].

Based on the above intuition, we take a fresh look at the IE problem and convert it to be a structured prediction task, where IE annotations are uniformly represented as *information networks*, and the goal of extraction becomes to in-

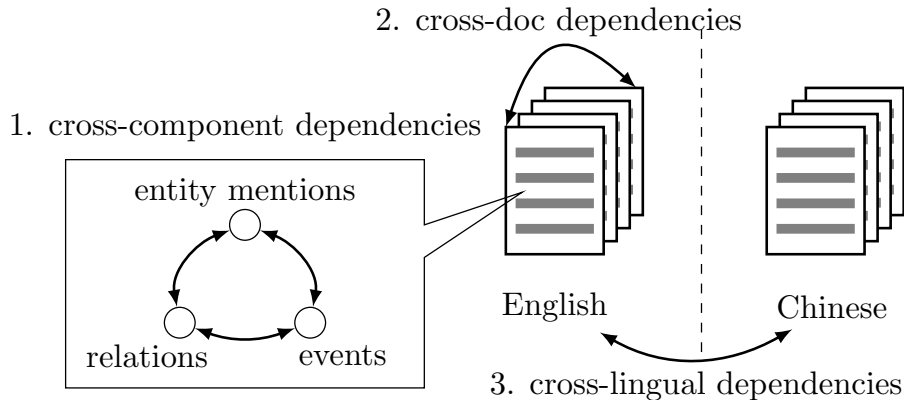


Figure 1.1: Overview of the main objectives of this thesis.

crementally extract the information networks from each sentence with local and non-local features. Using this framework, we can overcome the aforementioned limitations and capture: **(i) The interactions among multiple components.** Taking the sentence “*The tire maker still employs 1,400.*” as an example, we aim to extract “*tire maker*” as an **Organization (ORG)** entity mention, “*1,400*” as a **Person (PER)** entity mention, their **Employment-Organization (EMP-ORG)** relation, and the **Start-Position** event triggered by “*employ*”. Although it can be difficult for a mention extractor to extract “*1,400*” as a **Person (PER)** mention, the context word “*employs*” between “*tire maker*” and “*1,400*” strongly indicates an **Employment-Organization (EMP-ORG)** relation, which must involve a **PER** mention. Similarly, in the sentence “*From Michigan, Bush flies to Morgantown, West Virginia.*”, the **Physical (PHYS)** relation indicated by “*flies to*” between “*Bush*” and “*Morgantown*” can be used to infer “*Morgantown*” as a **Geopolitical Entity (GPE)** mention. **(ii) The global features of the entire structure.** The hidden IE structures often retain linguistic and logical constraints. And the local predictions are often dependent on one another. We can make use of global features or soft constraints to capture the dependencies of multiple nodes and edges in the output structures. For example, in the sentence “*US forces in Somalia, Haiti and Kosovo ...*”, we can design a global feature to ensure that the relations between “*forces*”, and each of the entity mentions “*Somalia*_{GPE}” and “*Kosovo*_{GPE}”, are of the same type (**Physical (PHYS)**, in this case). Experimental results on ACE corpora showed the advantage of this joint framework, and demonstrated the benefits

of performing global inference. The resultant model can jointly extract entity mentions, relations and events, and achieved state-of-the-art performance in each stage of the extraction.

We then go beyond sentence level to make use of cross-document dependencies. When we extract entities and their relations from a large corpus by using sentence-level extractors, there are often many erroneous and conflicting results. For example, when performing on 381,588 news documents from Global Autonomous Language Exploitation (GALE) corpora, a state-of-the-art single-document IE system produces more than 10 different incorrect country or region names to indicate where “*Osama Bin Laden*” was located. However, there exist a lot of dependencies among facts that are scattered from multiple documents. To improve the quality of extraction in a cross-document setting, we developed an Integer Linear Programming (ILP) based inference system to remove incorrect IE results by taking into account local confidence values, frequencies across documents, and a set of global constraints. For example, in a baseline result, “*George W. Bush*” may be detected as the member of both “*Republican Party*” with high confidence and “*Hamas*” with low confidence, while these two organizations are located in different regions (*United States* vs. *Palestine*). Based on one possible global constraint that an organization and its members are unlikely to locate in different regions, we can determine that “*George W. Bush*” is unlikely to be a member of “*Hamas*”.

Finally we break languages barriers by making use of cross-lingual dependencies. We consider name tagging as a case study to improve IE for parallel corpora. Effectively extracting and aligning names from bilingual data is important to various NLP and information access applications, such as named entity translation template [15], statistical word alignment [16], machine translation (MT) [17], cross-lingual IE (CLIE) [18] and many more. We argue that each language-specific tagger has its own advantages and disadvantages, and the features and resources from two languages are often complementary. For example, English person name tagging can utilize capitalization features while Chinese cannot; on the other hand Chinese person names are restricted to some certain characters while English translations are lack of this indicative feature [19]. We propose two novel bilingual name tagging

approaches based on conditional random fields (CRF) to make use of cross-lingual dependencies. The first approach is based on linear-chain CRF with cross-lingual features. The second approach jointly models each pair of sentences by introducing bilingual factors based on word alignment. Therefore the predictions from the two languages can mutually enhance each other to make more coherent predictions.

1.3 Related Publications

Some of the research work presented in this thesis has been published in the following peer-reviewed conference papers:

- Joint name tagging for bilingual corpora: [20,21]. [20] developed joint bilingual name tagging methods for parallel corpus. [21] applied the resulting tagger to improve name-aware machine translation.
- Cross-document inference for IE: [22]. [22] conducted cross-document information for relation and event predicates of entities based on an ILP formulation.
- Joint extraction of multiple IE elements: [23–25]. [23] developed the first joint model for ACE event trigger extraction and argument labeling. [24] applied segment-based decoding to perform joint extraction of entity mentions and relations. [25] summarized the formulation of *information networks*, and developed a system that combines [23] and [24]. To the best of our knowledge, this is the first work of jointly extracting the three fundamental IE components.

1.4 Thesis Structure

The remaining content of this thesis will be organized as follows: We will begin by overviewing the related work in the literature in Chapter 2. Chapter 3 presents the joint IE framework in detail, and describes the evaluation results on ACE’04 and ACE’05 corpora. Then we move to the cross-document inference in Section 4. Chapter 5 presents the algorithms for joint bilingual name tagging and its intrinsic and extrinsic evaluation results. Finally Chapter 6 will conclude this thesis and highlight some future research directions.

CHAPTER 2

Related Work

2.1 Entity Mention Extraction, Relation Extraction, and Event Extraction

IE techniques have advanced from rule-based to statistical and machine learning-based approaches. Rule-based methods use hand-coded patterns to extract information. While it is easy to implement and debug, they heavily rely on developers' heuristic and require lot of manual labor [26]. It usually has good precision but comparably low recall. Machine learning based approaches, on the other hand, are trainable, adaptable and extensible. With the development of human annotated corpora, machine learning based approaches have achieved significant progress.

Entity mention extraction and name tagging are commonly casted to sequential labeling problems in machine learning based methods. Nymble [27] is the first machine-learning based name tagger. It was trained from human-annotated corpus with a variant of Hidden Markov model (HMM), where each entity type is represented by a hidden state. Discriminative models such as Maximum Entropy Markov model (MEMM) and Conditional Random Fields (CRF) are then introduced to the task of name tagging and entity mention extraction [28–31]. Different from HMM, discriminative models directly model $p(y|x)$, the conditional distribution of label sequence y given the input sequence x . Therefore they can explore more expressive features based on the entire sentence. Most prior work only use local features.

Portions of this chapter previously appeared as: Q. Li, H. Li, H. Ji, W. Wang, J. Zheng, and F. Huang, "Joint Bilingual Name Tagging for Parallel Corpora," in *Proc. Int. Conf. on Inform. and Knowledge Manage.*, Maui, HI, 2012, pp. 1727–1731.

Portions of this chapter previously appeared as: Q. Li, H. Ji, and L. Huang, "Joint event extraction via structured prediction with global features," in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Sofia, Bulgaria, 2013, pp. 73–82.

Portions of this chapter previously appeared as: Q. Li and H. Ji, "Incremental joint extraction of entity mentions and relations," in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Baltimore, MA, 2014, pp. 402–412.

Portions of this chapter previously appeared as: Q. Li, H. Ji, Y. HONG, and S. Li, "Constructing information networks using one single model," in *Proc. Conf. on Empirical Methods in Natural Language Process.*, Doha, Qatar, 2014, pp. 1846–1851.

There are a few exceptions. For example, Finkel et al. [32] applied Gibbs sampling to incorporate non-local features. Krishnan et al. [33] introduced a two-stage system to make use of local decisions from the first stage.

Relation extraction is naturally considered to be a binary or multi-class classification problem. Some common classification algorithms have been used in this task. For example, [34] applied Maximum Entropy classifier (a.k.a. multi-class logistic regression) to ACE relation extraction. [11] used Support Vector Machine (SVM) and [35–38] studied kernel methods for this task. [39] systematically studied lexical, syntactic and semantic features for this task, [9] further explored various background knowledge, and [10] used syntactico-semantic information to improve relation extraction in ACE corpus.

Similar to relation extraction, event Extraction is usually defined as a series of binary or multi-class classification problems [40–42]. First, event trigger identification and classification is performed to extract event triggers and determine types. Then given a pair of previously extracted trigger and entity mention, argument identification and classification is applied to decide whether they have a argument relation and determine the argument role. Finally, an additional classifier may be employed to decide whether an extracted event mention is reportable or not [12]. SVM and Maximum Entropy classifiers are the most popular classification algorithms for these sub-tasks.

2.2 Utilizing Dependencies Across Languages

Some recent work has explored name tagging for parallel data. [43] presented a sequence of cost models to learn name translation pairs. This approach greatly relies on language-specific information such as repeated strings from both languages and capitalization clues. [44] proposed an approach to extract bilingual name pairs. Their method extracted names from each language first, and then computed the cost scores based on name tagging, name transliteration and word translation to rank candidate name pairs. [45] extended their ranking method by incorporating bilingual alignment, bilingual type re-assignment and monolingual candidate certainty. [46] described a joint inference model to improve entity extraction and translation.

All of these previous approaches can still be considered as adding a post-processing step after two isolated name taggers. In contrast, we develop a joint CRF approach to jointly perform name tagging on sentence pairs by using word alignment.

2.3 Joint Event Extraction

Most recent studies about ACE event extraction rely on staged pipeline that consists of separate local classifiers for trigger extraction and argument labeling [6–8, 12, 20, 40–42]. As far as we know, this thesis is the first attempt to jointly model these two ACE subtasks. For MUC and ProMed corpora, [47] proposed a probabilistic framework to extract event role fillers conditioned on the sentential event occurrence. Besides having different task definitions, the key difference from our approach is that the role filler recognizer and the sentential event recognizer are trained independently but combined in the test stage.

There has been some previous work on joint modeling for biomedical events [48–51]. [50] is most closely related to our work. They casted the problem of biomedical event extraction as a dependency parsing problem. The key assumption that event structure can be considered as trees is inapplicable in ACE event extraction. In addition, they used a separate classifier to predict the event triggers before applying the parser, while we extract the triggers and argument jointly. Moreover, the features in the parser are edge-factorized. To exploit global features, they applied a MaxEnt based global re-ranker. In comparison, the joint framework developed in this thesis is based on beam-search, which allows us to exploit arbitrary global features efficiently. In addition, our framework is also capable of extracting entity mentions jointly with event structures.

2.4 Joint Extraction of Entity Mentions and Relations

Entity mention extraction (e.g., [29, 52–55]) and relation extraction (e.g., [35–39, 56–60]) have drawn much attention in recent years but were usually studied separately. Most relation extraction work assumed that entity mention boundaries and/or types were given. [10] reported the best results in ACE corpus using system-predicted entity mentions.

Some previous work used relations and entity mentions to mutually enhance each other in joint inference frameworks, including re-ranking [61], Integer Linear Programming (ILP) [62–64], and Card-Pyramid Parsing [65]. [66] extended the ILP-based method to IQPs (Integer Quadratic Programs) formulation to better incorporate soft constraints. All these work noted the advantage of exploiting component interactions and using richer knowledge. But they used models separately learned for the two subtasks. In addition, the ILP-based framework can only exploit hand-coded hard constraints. In the IQPs formulation, penalty weights for soft constraints need to be tuned based on experiments in development set. As a key difference, our joint framework perform both joint training and decoding, and the weights for various global features can be learned during the training phase.

2.5 Other Joint Modeling Methods

The work of this thesis is largely motivated by the well-known research of constrained conditional models [62, 63, 67, 68], and also related to some other previous joint inference methods such as dual decomposition [69], and joint modeling methods based on probabilistic graph models (e.g., hierarchical conditional random fields [70, 71] and Markov logic networks [72–74]) which have been applied to various NLP tasks [20, 48, 49, 51, 62, 64, 65, 75, 76, 76–86]. Structured perceptron has been successfully used in other NLP tasks such as part-of-speech tagging and dependency parsing [87–89]. Although dependency parse tree can also be viewed as a graph structure, our task differs from dependency parsing in that structures in information extraction are more flexible, where each node can have arbitrary relation and/or event arcs. [75, 90, 91] used structured perceptron and beam-search for jointly predicting word segmentation, part-of-speech tagging, and dependency parsing. [92] generalized perceptron-like structured algorithms, and introduced the SEARN (shorthand of “search-learn”) framework. Based on whether the models for multiple tasks are *jointly* learned, or *separately* learned but combined by performing *joint inference* in the test phase, all of the aforementioned methods can be roughly categorized into two types:

- Joint Learning Algorithms. The model for multiple tasks is jointly learned.

Model parameters for the subtasks affect one and another during training. probabilistic graph models [70–74] and search-based models [75,90–92] belong to this category.

- Joint Inference Algorithms. The models for different tasks are separately learned. They are connected to each other by a joint inference component during the test. Dual decomposition [69] and constrained conditional models [62,63,67,68] fall into this category.

CHAPTER 3

Joint Information Extraction Framework with Structured Prediction and Inexact Search

In this chapter, we present the framework that jointly extracts multiple IE components within a sentence based on structured prediction and inexact search. We will show that in addition to extracting multiple facts simultaneously, the dependencies across local decisions can be captured by applying various non-local features.

3.1 Problem Definition and Motivation

The research problem that we are addressing here is the three fundamental tasks in information extraction: namely entity mention extraction, relation extraction, and event extraction. In the past years, each of them has been widely but separately studied. By contrast, we take a fresh look at those sub-tasks, and rise a research question: can the three subtasks be addressed by using a single joint model, and can the performance benefit from joint extraction?

Let us consider the entity mention and relation annotations depicted in Figure 3.1. From the sentence “*The tire maker still employs 1,400*”, we aim to extract “*tire maker*” as an **Organization (ORG)** entity mention, “*1,400*” as a “**Person (PER)**” entity mention, and their **Employment-Organization (EMP-ORG)** relation. A typical pipeline of end-to-end relation extraction consists of entity mention boundary identification, entity type classification and relation extraction. Similarly, a common pipeline of event extraction is composed of event trigger identification and classification, argument identification, and argument role classification. We argue

Portions of this chapter previously appeared as: Q. Li, H. Ji, and L. Huang, “Joint event extraction via structured prediction with global features,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Sofia, Bulgaria, 2013, pp. 73–82.

Portions of this chapter previously appeared as: Q. Li and H. Ji, “Incremental joint extraction of entity mentions and relations,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Baltimore, MA, 2014, pp. 402–412.

Portions of this chapter previously appeared as: Q. Li, H. Ji, Y. HONG, and S. Li, “Constructing information networks using one single model,” in *Proc. Conf. on Empirical Methods in Natural Language Process.*, Doha, Qatar, 2014, pp. 1846–1851.

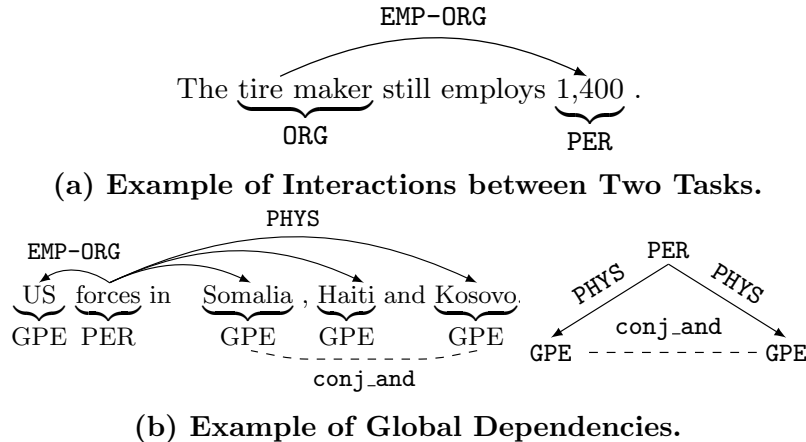


Figure 3.1: Examples of entity mention and relation extraction from ACE’04 corpus.

that such a pipelined approach has the following limitations: First, it prohibits the interactions between components. Errors in upstream components are propagated to the downstream classifiers, often resulting in compounding errors. The downstream components, however, cannot impact earlier decisions with feedback. Second, it over-simplifies the whole task as a set of multiple local classifiers without taking into account long-distance dependencies.

On the other hand, when skilled human readers distill information from documents, they do not predict different types of facts step by step in a “bottom-up” fashion; instead, they tend to take a “top-down” approach - attempting to comprehend content and predict high-level information through the lens of prior knowledge about the general subject before making sense of the details [13, 14]. Psycholinguistic research on incremental sentence processing (such as [93]) suggests that sentences are interpreted and disambiguated by human readers in an incremental fashion. Partial interpretations of what has been observed are made incrementally until the end of each sentence. Therefore, we convert the whole task to be a structured prediction task. By jointly extracting the information structures, we can break the aforementioned limitations by capturing:

1. The interactions among multiple tasks. Take Figure 3.1a as an example, although it could be difficult for a mention extractor to predict “1,400” as a Person (PER) mention, the context word “*employs*” between “*tire maker*”

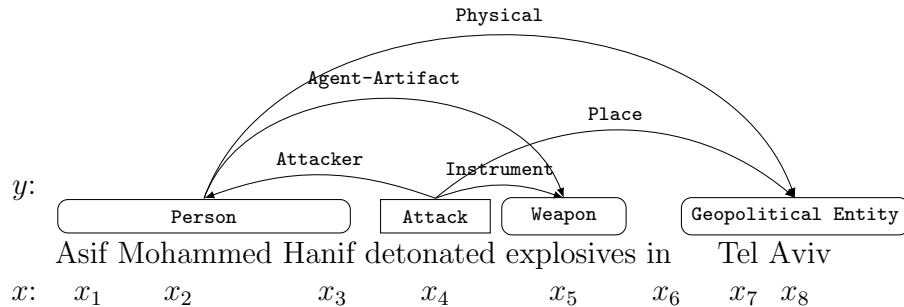


Figure 3.2: Information network representation. Information nodes are denoted by rectangles. Arrows represent information arcs.

and “1,400” strongly indicates a EMP-ORG relation which must involve a PER mention. Similarly, in Figure 3.1a, the Physical (PHYS) relation indicated by “flies to” between “Bush” and “Morgantown” can be used to infer “Morgantown” as a Geopolitical Entity (GPE) mention.

2. The global features of the graph structure. The hidden IE structures often retain linguistic and logical constraints. and local predictions are often dependent on one another. We can make use of global features or soft constraints to capture the dependencies of multiple nodes and edges in the output structures. For example, in Figure 3.1b, we can design the depicted graph feature to ensure that the relations between “forces”, and each of the entity mentions “Somalia_{GPE}” and “Kosovo_{GPE}”, are of the same type (Physical (PHYS), in this example).

Furthermore, we introduce a new representation for the task of information extraction. We formulate the IE output of each sentence as an *information network* $y(x) = (V, E)$, where V corresponds to *information nodes*, and E is the set of typed edges between each pair of nodes.

1. Information Node. Each node $v_i \in V$ is represented as a triple $\langle p_i, q_i, t_i \rangle$ of start index p_i , end index q_i , and node type t_i . A node can be an entity mention or an event trigger. A particular type of node is \perp (neither entity mention nor event trigger), whose maximum length is always 1.
2. Information Edge. Similarly, each information edge $e_j \in E$ is represented as

$\langle p_j, q_j, r_j \rangle$, where p_j and q_j are the end offsets of the nodes, and r_j is the edge type. An edge can either be a relation between a pair of entity mentions, or an argument link between an event trigger and an entity mention.

For instance, in Figure 3.2, the event trigger “*detonated*” is represented as $\langle 4, 4, \text{Attack} \rangle$, the entity mention “*Asif Mohammed Hanif*” is represented as $\langle 1, 3, \text{Person} \rangle$, and their argument edge is $\langle 4, 3, \text{Attacker} \rangle$. The goal of IE then becomes to extract the whole information network y for a given sentence x .

In this chapter, we use $\mathcal{L}_{\text{entity}} \cup \mathcal{L}_{\text{trigger}} \cup \{\perp\}$ to denote the node label alphabet, where $\mathcal{L}_{\text{entity}}$ represents the set of 7 main entity types, $\mathcal{L}_{\text{trigger}}$ is the 33 event subtypes, and \perp indicates that the token is not a trigger. Similarly, $\mathcal{R}_{\text{arg}} \cup \mathcal{R}_{\text{rel}} \cup \{\perp\}$ denotes the edge label sets, where \mathcal{R}_{rel} represents the set of directed relation types, \mathcal{R}_{arg} is the set of possible argument roles and directed relation types, and we override \perp to indicate that the pair of information nodes does not have any relation or argument link. The same relation type with different directions is considered as two types in \mathcal{R}_{rel} . For example, the binary relation “**Part-whole** ($\text{mention}_a, \text{mention}_b$)” and “**Part-whole** ($\text{mention}_a, \text{mention}_b$)” are two different relations.

In Section 3.2, we overview the baseline system that represents traditional pipelined approaches. Section 3.3 describes the framework that extracts *information networks* in detail.

3.2 Baseline Systems

In order to compare our proposed framework with traditional pipelined approaches, we developed a set of pipelined classifiers for extracting entity mentions, relations, and event mentions, respectively. All of the baseline classifiers are separately trained with the annotations for each subtask. In the test phase, these classifiers are cascaded together as a pipeline to predict each type of information sequentially. In our experiments on ACE corpora, they achieved comparable performance of state-of-the-art. Figure 3.3 demonstrates an overview of the pipelined architecture of the three IE sub-tasks. In order to obtain event arguments or relations, one has to classify boundaries and types of entity mentions beforehand.

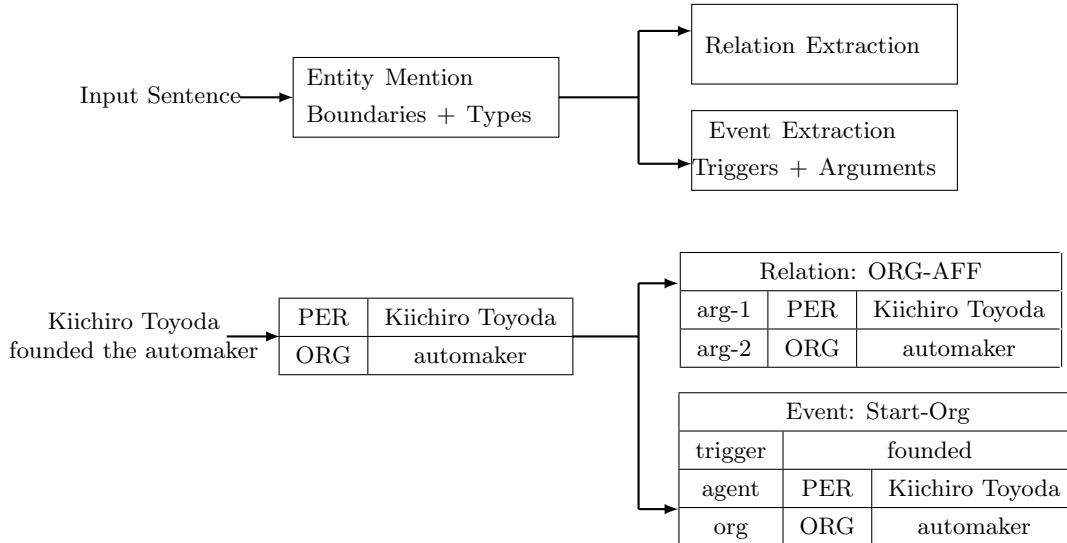


Figure 3.3: Overview of pipelined approach.

3.2.1 Entity Mention Classifier

We convert the problem of entity mention extraction to a sequential token tagging task as in the state-of-the-art system [52]. We applied the BILOU scheme, where each tag means a token is **B**eginning, **I**nside, **L**ast, **O**ut of, and **U**nit of an entity mention, respectively. As an example, the following table shows the BILOU tags for each token in the sentence in Figure 3.3:

y	Kiichiro	Toyoda	founded	the	automaker
x	B-PER	L-PER	O	O	U-ORG

Then we can train a linear-chain Conditional Random Fields (CRF) model [94] to predict the tag for each token. We use the Mallet [95] implementation of CRF in our experiments. Most of our features are similar to the work of [29, 52] except that we do not have their gazetteers and outputs from other mention detection systems as features. Our additional features are as follows:

- Governor word of the current token based on dependency parsing [96].
- Prefix of each word in Brown clusters learned from TDT5 corpus [56].

3.2.2 Relation Classifier

Given a sentence with entity mention annotations, the goal of the baseline relation extraction is to classify each pair of mentions into one of the pre-defined relation types with direction or \perp (non-relation). Theoretically we can employ any popular multi-class classification algorithm to classify the relations, among which Support Vector Machine (SVM) [97] and Maximum Entropy (MaxEnt) model are usually used in the prior work for relation extraction and often yield comparable performance. We chose MaxEnt model and its implementation in Mallet for our experiments. Most of our relation extraction features are based on the previous work of [11] and [34]. We designed the following additional features:

- Sequence of phrase labels for the sub-sentence covering the two mentions. For example, for the sentence in Figure 3.1a, the sequence is “NP, VP, NP”. We also augment it by head words of each phrase.
- Four syntactico - semantic patterns described in [9].
- We replicated each lexical feature by replacing each word with its Brown cluster prefix [56].

3.2.3 Event Mention Pipeline

Most state-of-the-art approaches [6–8] used sequential pipelines as building blocks for event mention extraction, which break down the whole task into many separate subtasks, such as trigger identification/classification and argument identification/classification. Chen and Ng [42] have proven that performing identification and classification in one step is better than two steps. We implemented two MaxEnt classifiers for trigger labeling and argument labeling respectively. The detailed description about the features is summarized in Figure 3.1 and Figure 3.2.

3.3 Joint Extraction Framework

In this chapter, we aim to extract information networks of multiple IE components via joint structured prediction, along with arbitrary global features estimated from training stage. In this way, various IE annotations can be coherently extracted

Table 3.1: Features in event trigger detection.

Category	Feature Description
Lexical	<ol style="list-style-type: none"> 1. unigrams/bigrams of the current and context words within the window of size 2 2. unigrams/bigrams of pos tags of the current and context words within the window of size 2 3. lemma and synonyms of the current token 4. base form of the current token extracted from Nomlex [98] 5. Brown clusters of the current token [99]
Syntactic	<ol style="list-style-type: none"> 6. dependent and governor words of the current token 7. dependency types associated the current token 8. whether the current token is a modifier of job title 9. whether the current token is a non-referential pronoun
Entity Information	<ol style="list-style-type: none"> 10. unigrams/bigrams normalized by entity types 11. dependency features normalized by entity types 12. nearest entity type and string in the sentence or clause

Table 3.2: Features in event argument labeling.

Category	Feature Description
Basic	<ol style="list-style-type: none"> 1. context words of the entity mention 2. trigger word and type 3. entity type, subtype and entity role if it is a geo-political entity mention 4. entity mention head, and head of any other name mention from co-reference chain 5. lexical distance between the argument candidate and the trigger 6. the relative position between the argument candidate and the trigger: {before, after, overlap, or separated by punctuation} 7. whether it is the nearest argument candidate with the same type 8. whether it is the only mention of the same entity type in the sentence
Syntactic	<ol style="list-style-type: none"> 9. dependency path between the argument candidate and the trigger 10. path from the argument candidate and the trigger in parse tree 11. length of the path between the argument candidate and the trigger in dependency graph 12. common root node and its depth of the argument candidate and parse tree 13. whether the argument candidate and the trigger appear in the same clause

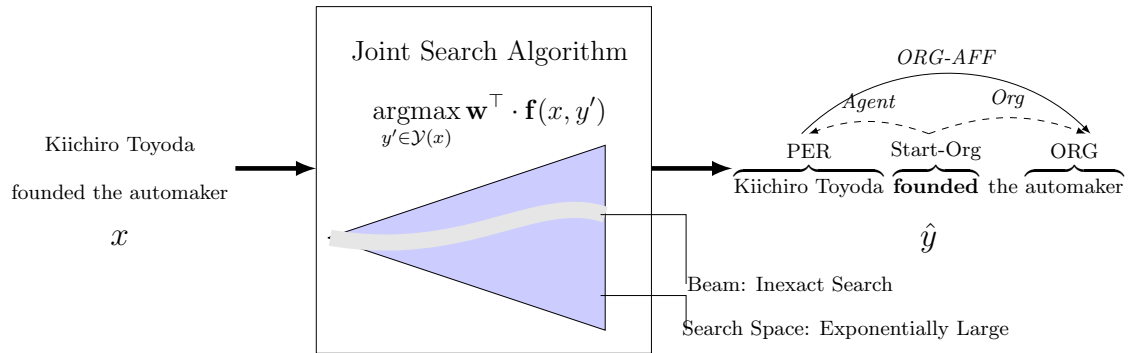


Figure 3.4: Overview of joint extraction framework.

at the same time in a joint search space. Figure 3.4 illustrates an overview of the framework. Taking an English sentence x as input, the framework employs beam search to efficiently search for the best hypothesis of information network \hat{y} under the model parameters \mathbf{w} . An *information network* can include multiple types of annotations, such as entity mentions (e.g., PER mention “Kiichiro Toyoda” and ORG mention “automaker”), relations (e.g., ORG-AFF relation), and events (e.g., Start-Org event triggered by “founded”). Different from the baseline systems depicted in Figure 3.3, those annotations are extracted jointly with a single model. In this section, we describe the training and decoding methods for the joint framework in detail.

3.3.1 Structured Perceptron with Beam Search

We apply structured perceptron to estimate model parameters \mathbf{w} from training data. Structured perceptron, proposed in [87], is an extension to the standard linear perceptron for structured prediction. It has been successfully applied in many other Natural Language Processing (NLP) tasks, such as parsing [87, 89], part-of-speech tagging [91], and word segmentation [75, 90]. Given an input instance $x \in \mathcal{X}$, which in IE tasks can often be a sentence with partial annotations, the structured perceptron finds the best configuration of the structure $z \in \mathcal{Y}(x)$ by the following linear model:

$$z = \underset{z' \in \mathcal{Y}(x)}{\operatorname{argmax}} \mathbf{w}^\top \cdot \mathbf{f}(x, z') \quad (3.1)$$

where $\mathbf{f}(x, z)$ represents the feature vector that characterizes configuration z for instance x , and \mathbf{w} is the corresponding weights vector. We will show that we not

only use local features, but also exploit a variety of global features in our tasks.

In order to estimate the weight parameters in the model, the algorithm applies an on-line updating schedule. Let $\mathcal{D} = \{(x^{(j)}, y^{(j)})\}_{j=1}^n$ be the set of training instances (where i denotes the index of the current training instance). In each iteration, the algorithm uses the linear function defined in Eq. (3.1) to search for the best configuration z for the input sentence x under the current parameters. If z is incorrect, then the parameters are updated such that the features of the current instance are moved towards the gold-standard y , and against z :

$$\mathbf{w}^{new} = \mathbf{w} + \mathbf{f}(x, y) - \mathbf{f}(x, z) \quad (3.2)$$

In addition to the simple perceptron update, we also apply k-best MIRA method [100], an online large-margin learning algorithm. During each update, it keeps the norm of the change to feature weights \mathbf{w} as small as possible, and forces the margin between y and the k-best candidate z greater or equal to their loss $L(y, z)$. It can be formulated as a quadratic programming problem:

$$\begin{aligned} & \text{minimize} && \|\mathbf{w}_{new} - \mathbf{w}\| && (3.3) \\ & \text{subject to} && \mathbf{w}_{new}^\top \cdot (\mathbf{f}(x, y) - \mathbf{f}(x, z)) \geq L(y, z) \\ & && \forall z \in \text{best}_k(x, \mathbf{w}) \end{aligned}$$

We use coordinate descent algorithm to solve it. Comparing with perceptron update, k-best MIRA has several advantages: it is flexible in using various loss functions, it is a large-margin approach, and can use multiple candidate structures to tune feature weights.

The key step of the training and test is the decoding procedure, which aims to search for the best configuration under the current parameters. In simpler tasks such as part-of-speech tagging and noun phrase chunking, efficient dynamic programming algorithms (such as Viterbi algorithm) can be employed to perform exact inference. Unfortunately, it is intractable to perform the exact search in our framework because:

1. the search space becomes much more complex when multiple IE components

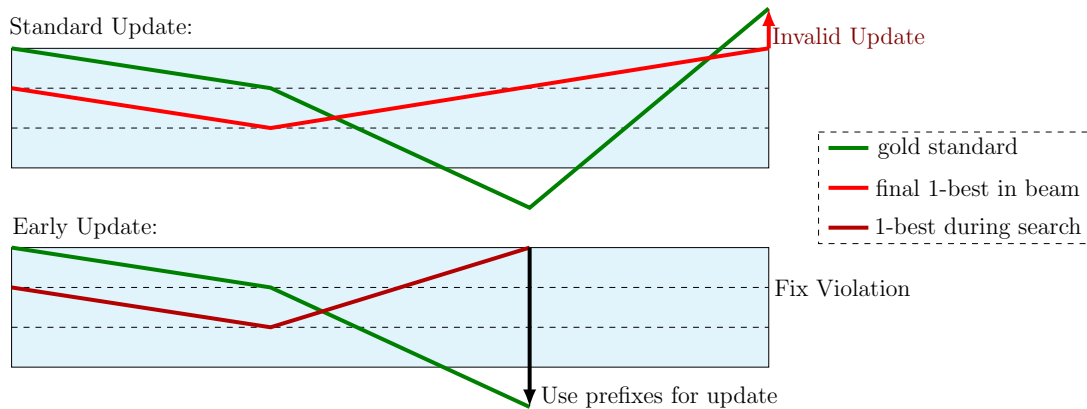


Figure 3.5: Example of early update in beam search. In this toy example, each rectangle represents the beam search that maintains three candidates at each step.

are jointly extracted.

2. we propose to make use of arbitrary global features, which makes it infeasible to perform exact inference efficiently.

As a tradeoff between optimality and efficiency, in this framework we employ beam-search, an instance of inexact search, to approximate Eq.3.1. Since the search is bounded by the beam size, and the final top candidate is not guaranteed to be globally optimal, the original learning procedure of perceptron may lead to invalid updates. To avoid this problem we adopt the early-update mechanism for training. Early-update mechanism was first introduced in [88], and [101] later proved the convergence property and formalized a general framework named violation-fixing perceptron. Figure 3.5 demonstrates the difference between the standard update and early update in beam search. In the standard update, the gold-standard assignment y may fall out of the beam at some point during the search, but its final model score can be higher than the model score of the top candidate z in the beam. This leads to an invalid update [101]. Early update, on the other hand, stops searching when the gold-standard y falls out of the beam, and uses prefix configurations for update. This strategy forms a violation fixing update, and leads less noisy to the parameter estimation compared to standard update.

Figure 10 describes the skeleton of the perceptron training algorithm with

Input: training set $\mathcal{D} = \{(x^{(j)}, y^{(j)})\}_{j=1}^N$, maximum iteration number T
Output: model parameters \mathbf{w}

```

1 initialize  $\mathbf{w}, \mathbf{w}_a \leftarrow \mathbf{0}$ 
2  $c \leftarrow 1$ 
3 for  $t \leftarrow 1 \dots T$  do
4   foreach  $(x, y) \in \mathcal{D}$  do
5      $(x, y', z) \leftarrow \text{BEAMSEARCH}(x, y, \mathbf{w})$ 
6     if  $z \neq y$  then
7        $\mathbf{w} \leftarrow \mathbf{w} + \mathbf{f}(x, y') - \mathbf{f}(x, z)$ 
8        $\mathbf{w}_a \leftarrow \mathbf{w}_a + c(\mathbf{f}(x, y') - \mathbf{f}(x, z))$ 
9        $c \leftarrow c + 1$ 
10 return  $\mathbf{w} \leftarrow \mathbf{w} - \mathbf{w}_a / c$ 

```

Figure 3.6: Perceptron learning algorithm with beam-search and early-update. y' is the prefix of the gold-standard and z is the top assignment in beam.

beam search. The update function can be replaced by k-best MIRA method in E.q.(3.3). Finally, to reduce overfitting, we use averaged parameters after training to decode test instances in our experiments. The resulting model is called averaged perceptron [87, 102]. Let \mathbf{w}_j^t be the weight vector from t -th ($t = 1 \dots T$) iteration and j -th ($j = 1 \dots N$) instance during the training. The final averaged weight vector is $(\sum_{j,t} \mathbf{w}_j^t) / (N \cdot T)$. It is very expensive to compute it directly, therefore we implemented an efficient method described in [92] to calculate the averaged vector. As shown in Figure 10, at each step, the algorithm maintains the original weight vector \mathbf{w} , as well as a bias vector \mathbf{w}_a and a decay variable c . And the final averaged parameter vector equals $\mathbf{w} - \mathbf{w}_a / c$.

Given this learning algorithm, the remaining challenge of developing a joint extraction framework is to design an efficient and effective search algorithm, and exploit informative features to capture the patterns of the hidden structures.

3.3.2 Decoding Algorithms

Based on the formulation of *Information Networks*, in general there are two types of actions during the decoding:

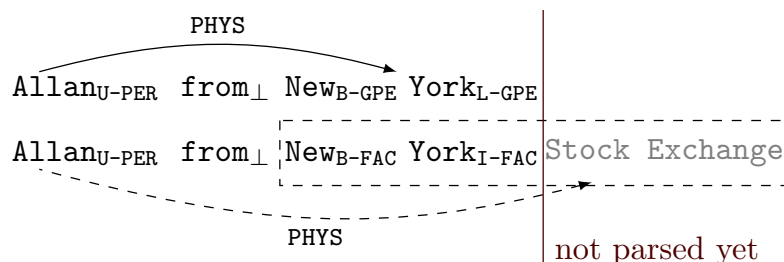
1. Node Step. Search for information nodes such as entity mentions and event

triggers.

2. Edge Step. Search for typed edges between each pair of nodes, such as relations and event argument links.

During the beam search, it is necessary to synchronize all candidates at each step so that they can be evaluated and ranked fairly. In the setting of name tagging, the decoding problem is commonly considered to be a sequential labeling task with BIO or BILOU scheme [31,52], where each sentence is regarded as a sequence of tokens. In this chapter, we are aiming to go beyond name tagging and jointly perform decoding for multiple sub IE tasks. As a result, it cannot be simply approached by sequential labeling. However, the input, namely a sentence, can still be viewed as a sequence of tokens. Hence, one straightforward decoding strategy is to extend the sequential labeling schema. From left to right, at each step of the search, the algorithm absorbs a token, makes analysis of the prefix, and propose partial structures (such as entity type, relation, event trigger etc.) to the token. We refer to this strategy as token-based decoding algorithm as the search is synchronized by token indices.

The token-based method is simple but has some limitations. Consider the task of jointly extracting entity mentions and their relations. If at each step, we assign a token with one of BIO or BILOU labels, it is unfair to compare the model scores of a partial mention and a complete mention during beam search. It is even more difficult to synchronize the search process of relations. For example, consider the two hypotheses ending at “York” for the same input sentence “Allen from New York Stock Exchange”:



Assuming the algorithm is processing the token “York”, and the rest of the sentence has not been parsed yet in both cases. The model would bias towards the incorrect

Table 3.3: Comparison between token-based decoding and segment-based decoding with examples of entity mention extraction.

Method	Example on Entity Mention Extraction	Time Complexity
Token-based (Florian et. al., 2006) (Ratinov & Roth, 2009)	The tire maker still employs 1,400 ⊥ B-ORG L-ORG ⊥ ⊥ U-PER	$O(n)$ n is number of tokens.
Segment-based (Sarawagi & Cohen, 2004) (Zhang & Clark, 2008)	The tire maker still employs 1,400 ┌───┐ ┌───┐ ORG PER	$O(n \cdot d)$ d is the maximal length of entity mentions.

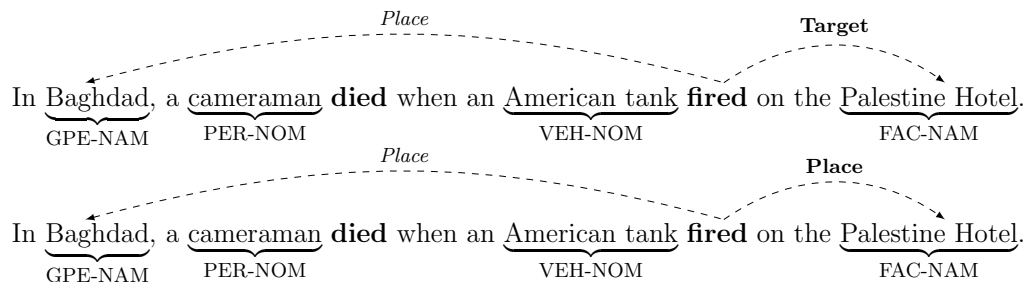
assignment “ $New_{/B-GPE} York_{/L-GPE}$ ” since it can have more informative features as a complete mention (e.g., a binary feature indicating if the entire mention appears in a GPE gazetteer). Furthermore, the predictions of the two PHYS relations cannot be synchronized since “ $New_{/B-FAC} York_{/I-FAC}$ ” is not yet a complete mention. To tackle these problems, we employ the idea of semi-Markov chain [103], in which each state corresponds to a segment of the input sequence. They presented a variant of Viterbi algorithm for exact inference in semi-Markov chain. We relax the max operation by beam-search, resulting in a segment-based decoder similar to the multiple-beam algorithm in [75]. Let \hat{d} be the maximum length of information nodes. The k -best partial assignments ending at the i -th token can be calculated as:

$$B[i] = \underset{z' \in \{z_{[1..i]} | z_{[1..i-d]} \in B[i-d], d=1..d\}}{\text{best}_k} \mathbf{w}^\top \cdot \mathbf{f}(x, z')$$

where $z_{[1..i-d]}$ stands for a partial configuration ending at the $(i-d)$ -th token, and $z_{[i-d+1..i]}$ stands for an information node over the new segment from index $(i-d+1)$ to i . We refer to this method as segment-based decoding algorithm. Table 3.3 shows comparisons between the token-based and the segment-based methods. We can see that the segment-based decoding has greater time complexity. And when d equals 1, it is identical to the token-based method.

3.3.3 Global Feature vs. Local Feature

One novelty of our framework is the use of global features to capture long-distance and cross-component dependencies. Traditional pipelined approaches to information extraction can only use local features to characterize single units in the output structures. For example, an event argument classifier classifies the type of each argument instance independently, therefore it is only based on features of each individual instance. Making decisions on one candidate does not affect another. For entity mention detection, linear-chain Markov model such as CRF is commonly used, where the distribution of each sentence is factorized to a sequence of local factors, therefore it is difficult to exploit long-distance dependencies. By contrast, we propose to exploit arbitrary global features to flexibly capture the properties of the entire output structure. In this way, local decisions can affect each other through the information captured by global features. Here we define the global features as the features involve multiple local decisions, which are independently made by multiple local classifiers in pipelined approaches. For example, consider the two hypotheses of **Attack** event triggered by “*fired*” in the same sentence:



The first structure is correct, while the second one mistakenly labels “*Palestine Hotel*” as a **Place** argument to “*fired*”. It is unusual that one event trigger can have multiple **Place** arguments. By encoding this common knowledge as a global feature, we can promote the structure that has only one **Place** argument, or penalize the one with multiple **Place** arguments. Although adding arbitrary global features makes the global inference intractable, inexact inference methods, such as beam-search, make it feasible while sacrificing optimality. In the experiments we will show that, in practice, beam-search can work very well.

3.3.4 Entity Type Constraints

Table 3.4: Examples of entity-type constraints in ACE’05 corpus.

Event/Relation Type	Argument Role	Entity Types
Transport	Agent	PER ORG GPE
	Artifact	PER WEA VEH
	Vehicle	VEH
Be-Born	Person	PER
	Place	GPE LOC FAC
Attack	Attacker	PER ORG GPE
	Target	PER ORG VEH FAC WEA LOC
	Instrument	WEA VEH
	Place	GPE LOC FAC
PHYS	Arg-1	PER FAC GPE LOC
	Arg-2	FAC LOC GPE
PER-SOC	Arg-1	PER
	Arg-2	PER

Entity type constraints have been shown useful in predicting relations [9, 63]. For instance, according to the ACE’05 annotation guideline, only **PER** entity can participate in **PER-SOC** relation. Similarly, the event structures also must follow a certain constraints. Some event arguments are general to all kinds of events, such as place and time arguments. Most arguments, on the other hand, are event-specific. For example, **Be-Born** events have **Person** argument slots, while **Attack** events have **Attacker**, **Instrument**, and **Target** slots. In addition, each type of argument slot can only be filled by entity mentions with some particular types. For example, **Person** arguments can only be fulfilled by **PER** mentions, while **Attacker** can be **PER**, **ORG**, or **GPE** mentions. Table 3.4 shows some typical examples of entity type constraints in ACE’05 corpus.

Instead of applying the constraints in post-processing inference, we can prune the branches that violate type constraints during the search, so that the constraints can be applied in both training and test phases. The pruning can largely reduce the search space as well as make the input for parameter update less noisy. If the entity type constraints are clearly defined by the task definition, we can manually create a frame that resembles Table 3.4. In the case that there does not exist any clear type constraints in the task definition, we can collect these information automatically from training data.

3.4 Token-based Decoding

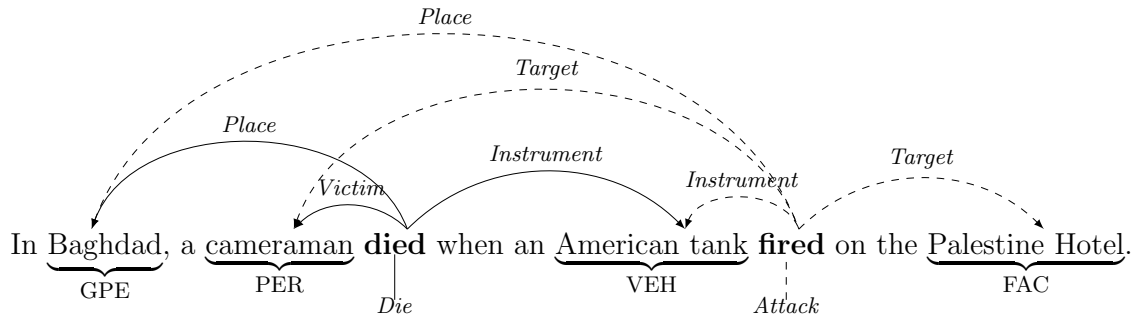


Figure 3.7: Example of event mentions. There are two event mentions that share three arguments: the Die event mention triggered by “died”, and the Attack event mention triggered by “fired”.

Given an English sentence with and argument candidates such as ACE entity mention, value, and temporal expression annotations, we can apply the token-based decoding algorithm to extract the event triggers and their arguments jointly. Figure 3.7 depicts an example that contains two event mentions and five entity mentions. More formally, let $x = \langle (x_1, x_2, \dots, x_s), \mathcal{E} \rangle$ denote the sentence instance, where x_i represents the i -th token in the sentence and $\mathcal{E} = \{e_k\}_{k=1}^m$ is the set of argument candidates (such as entity mentions, values, and temporal expressions). The goal of joint event extraction is to predict the corresponding hidden event structure from x :

$$y = (t_1, a_{1,1}, \dots, a_{1,m}, \dots, t_s, a_{s,1}, \dots, a_{s,m})$$

where t_i represents the trigger assignment for the token x_i , and $a_{i,k}$ represents the argument role label for the edge between x_i and argument candidate e_k .

$$\begin{array}{cccc}
 g(1) & g(2) & h(2, 1) & h(3, 2) \\
 \downarrow & \downarrow & \downarrow & \downarrow \\
 y = (t_1, a_{1,1}, a_{1,2}, t_2, \underbrace{a_{2,1}, a_{2,2}}_{\text{arguments for } x_2}, t_3, a_{3,1}, a_{3,2})
 \end{array}$$

Figure 3.8: Example notation with $s = 3, m = 2$.

For simplicity, throughout this chapter we use $y_{g(i)}$ and $y_{h(i,k)}$ to represent t_i and $a_{i,k}$, respectively. Figure 3.8 demonstrates the notation with $s = 3$ and $m = 2$. The

Input: Instance $x = \langle (x_1, x_2, \dots, x_s), \mathcal{E} \rangle$ and the oracle output y if for training.

k : Beam size.

$\mathcal{L}_{\text{trigger}} \cup \{\perp\}$: trigger label alphabet.

$\mathcal{R}_{\text{arg}} \cup \{\perp\}$: argument role alphabet.

Output: Top prediction z for x

```

1 Set beam  $B \leftarrow [\epsilon]$  /* empty configuration */
2 for  $i \leftarrow 1 \dots s$  do
3    $buf \leftarrow \{z \circ l \mid z \in B, l \in \mathcal{L}_{\text{trigger}} \cup \{\perp\}\}$   $B \leftarrow \text{k-best}(buf)$ 
4   if  $y_{[1:g(i)]} \notin B$  then
5     return  $B[0]$  /* for early-update */
6   for  $e_k \in \mathcal{E}$  do
7      $buf \leftarrow \emptyset$  /* search for arguments */
8     for  $z \in B$  do
9        $buf \leftarrow buf \cup \{z \circ \perp\}$ 
10      if  $z_{g(i)} \neq \perp$  then
11         $buf \leftarrow buf \cup \{z \circ r \mid r \in \mathcal{R}_{\text{arg}}\}$  /*  $x_i$  is a trigger */
12       $B \leftarrow \text{k-best}(buf)$ 
13      if  $y_{[1:h(i,k)]} \notin B$  then
14        return  $B[0]$  /* for early-update */
15 return  $B[0]$ 

```

Figure 3.9: Token-based decoding algorithm. $z \circ l$ means appending label l to the end of z . During test, lines 4-5 & 13-14 are omitted.

variables for the toy sentence “Jobs founded Apple” are as follows:

$$\begin{aligned}
 x &= \langle (Jobs, \overbrace{founded}^{x_2}, Apple), \overbrace{\{Jobs_{\text{PER}}, Apple_{\text{ORG}}\}}^{\mathcal{E}} \rangle \\
 y &= (\perp, \perp, \perp, \underbrace{Start\text{-}Org}_{t_2}, \underbrace{Agent, Org}_{\text{args for } founded}, \perp, \perp, \perp)
 \end{aligned}$$

Typical event extraction systems, such as [6, 7], take each token as a unit to identify and classify event triggers. Based on this strategy, we design a token-based decoding algorithm to jointly search for event triggers and their arguments. The search process is indexed by the token number. In each step, the algorithm absorbs a token in order, make event type hypotheses about this token, and then classifies the relation between argument candidates and the newly identified trigger in a left-to-right manner. Figure 3.9 describes the decoder with early-update.

There are two types of actions at each token:

1. Node (Trigger) Action: (line 3) Assign trigger labels for the current token. The linear model defined in Eq. (3.1) is used to score each partial configuration. Then the *k-best* partial configurations are selected to the beam.
2. Edge (Argument) Action: (line 11) Link arguments to new trigger. Once a trigger label for x_i is found in the beam, the decoder searches through the set of argument candidates \mathcal{E} to label the edges between each argument candidate and the trigger. The assignments are re-ranked after labeling each argument candidate, and *k-best* results are selected to the beam.

There are s steps of node actions in total for a sentence. And each of them is followed by m steps of edge actions. It is easy to show that the overall decoding time complexity is $O(k \cdot s \cdot m)$.

3.4.1 Decoding Example

Taking a simplified version of the sentence in Figure 3.7 as an example, Figure 3.10 illustrates several steps of the decoding algorithm, where the blue lines and words indicate the current step, and the red words indicate the correct types in each step. For clarity, it only shows one entry in the beam at each step. There are 5 tokens and 2 entity mentions, therefore $s = 5$ and $m = 2$. At Figure 3.10a, we assume that the index $i = 6$, and the algorithm is making prediction about the trigger label of “died”. There are multiple hypothesis, among which **Die** is the correct assignment. After setting **Die** as trigger label for “died”, it continues to predict the edge label between “died” and the **PER** mention “cameraman” (Figure 3.10b). Among other candidates, **Victim** is the correct assignment. Next, the edge label between “died” and the **GPE** mention “Baghdad” is determined from all possible candidates (Figure 3.10c). Lastly, the final structure is obtained.

3.5 Segment-based Decoding

Our final goal is to automatically extract entity mentions, relations and events from raw texts. More formally, let $x = (x_1, x_2, \dots, x_s)$ be a sentence instance, where x_i represents the i -th token. The goal is to extract y , the annotations of entity

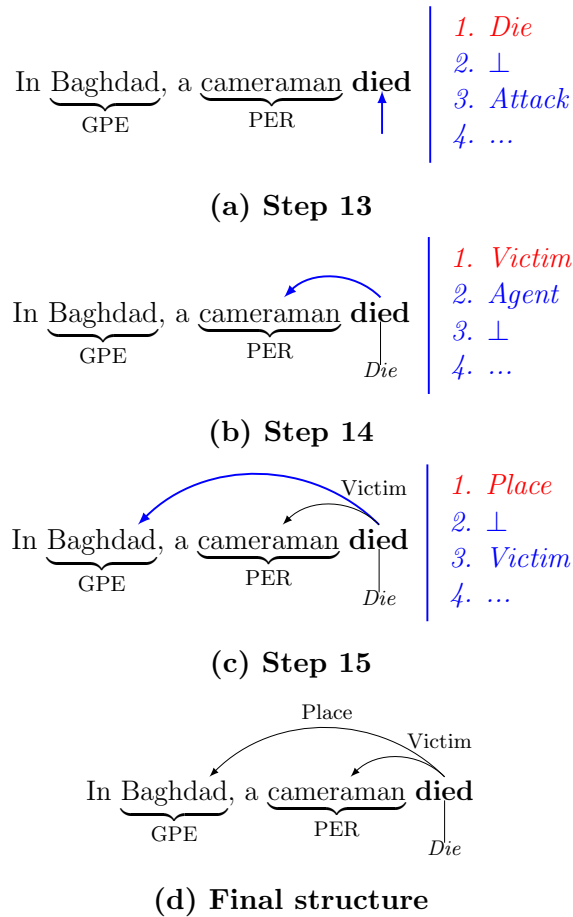


Figure 3.10: Example of decoding steps.

mentions, relations, and events for x . In our formulation of *information networks*, the entity mentions or event triggers in the output structure y can be expressed as a list of nodes (v_1, \dots, v_m) , where each segment $v_i = \langle p_i, q_i, t_i \rangle$ is a triple of start index p_i , end index q_i , and node type t_i . And each relation or event argument link e_j is an information edge between a pair of entity mentions, and can be denoted by $\langle p_j, q_j, r_j \rangle$, where p_j and q_j are the end offsets of its nodes, and r_j is the edge type (relation type or event argument role). Figure 3.2 in Section 3.1 gave an example about the notation of the information networks, we skip concrete examples here.

We employ the segment-based decoding algorithm to incorporate entity mentions and relations into the joint framework. Let \hat{d} be the upper bound of entity mention length. The joint decoding algorithm is shown in Figure 3.11. For each token index i , it maintains a beam for the partial assignments whose last segments

Input: input sentence $x = (x_1, x_2, \dots, x_s)$.
 k : beam size.
 $\mathcal{L}_{\text{entity}} \cup \mathcal{L}_{\text{trigger}} \cup \{\perp\}$: node type alphabet.
 $\mathcal{R}_{\text{rel}} \cup \mathcal{R}_{\text{arg}} \cup \{\perp\}$: edge type alphabet.⁵
 d_t : max length of type- t segment, $t \in \mathcal{L}_{\text{entity}} \cup \mathcal{L}_{\text{trigger}}$.
Output: best configuration z for x

```

1 initialize  $m$  empty beams  $B[1..m]$ 
2 for  $i \leftarrow 1..m$  do
3   for  $t \in \mathcal{L}_{\text{entity}} \cup \mathcal{L}_{\text{trigger}} \cup \{\perp\}$  do
4     for  $d \leftarrow 1..d_t, y' \in B[i-d]$  do
5        $q \leftarrow i-d+1$ 
6          $B[i] \leftarrow B[i] \cup \text{APPEND}(y', t, q, i)$  /* start offset */
7    $B[i] \leftarrow \text{best}_k(B[i])$  /* append new segment */
8   for  $j \leftarrow (i-1)..1$  do
9      $\text{buf} \leftarrow \emptyset$ 
10    for  $z' \in B[j]$  do
11      if  $\text{HASPAIR}(z', i, j)$  then
12        for  $r \in \mathcal{R}_{\text{rel}} \cup \mathcal{R}_{\text{arg}} \cup \{\perp\}$  do
13           $\text{buf} \leftarrow \text{buf} \cup \text{LINK}(z', r, i, j)$  /* create new edge */
14        else
15           $\text{buf} \leftarrow \text{buf} \cup \{z'\}$ 
16       $B[i] \leftarrow \text{best}_k(\text{buf})$ 
17 return  $B[m][0]$ 

```

Figure 3.11: Segment-based decoding algorithm. $\text{HASPAIR}(z', i, j)$ checks if there are two information nodes in z' that end at token x_i and token x_j , respectively. $\text{APPEND}(z', t, k, i)$ appends z' with a type- t segment spanning from x_k to x_i . Similarly $\text{LINK}(z', r, i, j)$ creates a new assignment from z' by assigning a directed edge with type r to the pair of nodes ending at x_i and x_j respectively.

end at the i -th token. Therefore the search is synchronized by the last token of each segment. There are two types of actions during the search:

1. Node Action (Lines 3-7). First, the algorithm enumerates all possible segments (i.e., subsequences) of x ending at the current token with various node types. A special type of segment is a single token with negative label (\perp). Each segment is then appended to the existing partial assignments in one of the previous beams to form new assignments. Finally the top k results are recorded in the current

beam.

- Edge Action (Lines 8-16). After each step of node action, the algorithm looks backward to link the newly identified information nodes and previous ones (if any) with information edges. At the j -th sub-step, it only considers the previous mention ending at the j -th previous token. Therefore different configurations are guaranteed to have the same number of sub-steps. Finally, all assignments are re-ranked with the new edge information.

The maximum length of each type of node \hat{d}_t is collected from the training data at the beginning of the training procedure. The following table summarizes the maximum length of each type of node in our experiments. We can see that the numbers that we collected from our training data are larger than or equal to those in our test data. Among others, ORG has the longest mentions such as “*Pearl River Hang Cheong Real Estate Consultants Ltd*”.

Node Type	Max Length		Node Type	Max Length	
	Train	Test		Train	Test
Event or \perp	1	1	GPE	5	4
PER	6	6	ORG	8	6
WEA	4	2	VEH	4	4
LOC	3	2	FAC	6	5

For entity type constraints, we automatically collected a mapping table of permissible entity types for each relation type from our training data. In our experiments, only 7 relation mentions (0.5%) in the development set and 5 relation mentions (0.3%) in the test set violate the constraints collected from the training data. As for events, the manual annotation in the ACE’05 corpus strictly obeys entity type constraints declared in the official annotation guideline. Therefore we simply created a mapping table based on the official annotation guideline.

There are m steps of node actions. The inner loop of node action executes at most $\hat{d} \cdot k$ times. At each token x_i , $i - 1$ times of edge action are performed. As such, the worst-case time complexity of this algorithm is $O(\hat{d} \cdot k \cdot s^2)$, where \hat{d}

⁵The same relation type with opposite directions is considered to be two classes in \mathcal{R} .

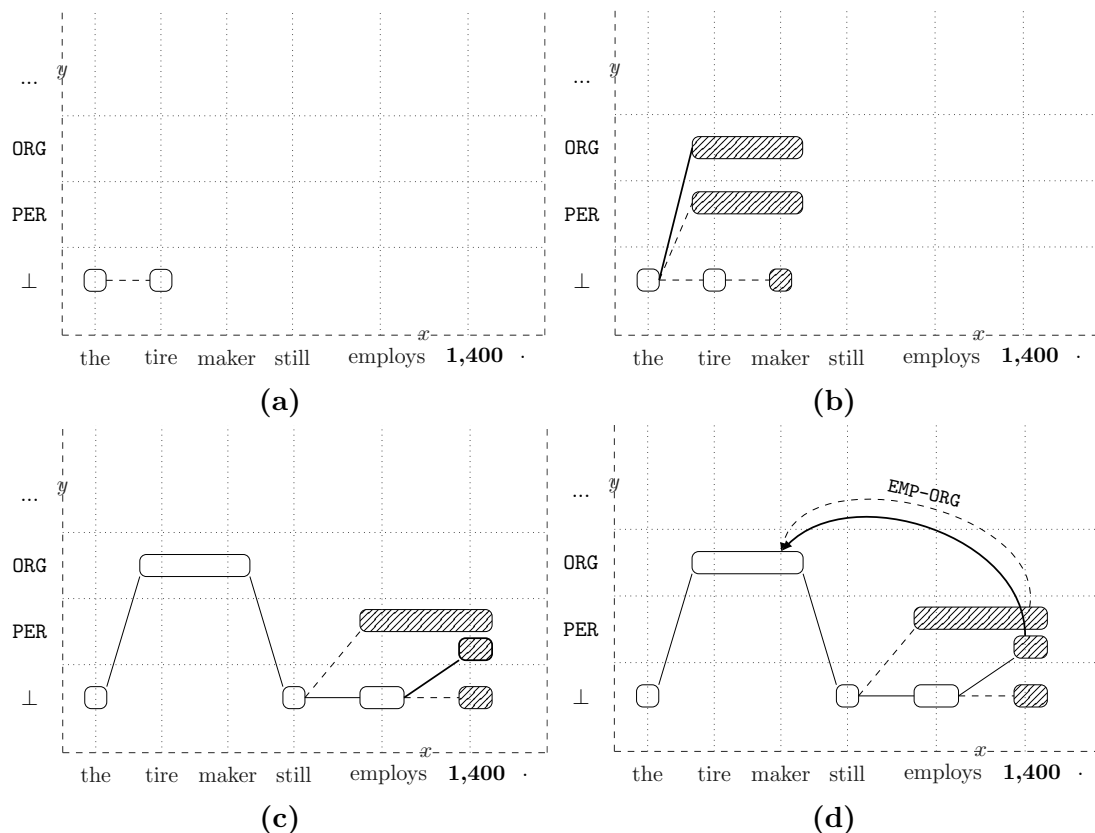


Figure 3.12: Example of decoding steps. x -axis and y -axis represent the input sentence and target node types, respectively.

is the upper bound of segment length. It is worth noting that this framework can be viewed as a generalized version of the algorithm described in Figure 3.9. If each segment’s length is 1, the process assembles the token-based decoder.

3.5.1 Decoding Example

Comparing with the token-based decoder in the previous section, the segment-based decoding is more complex but flexible. Here we demonstrate a simple but concrete example by considering again the sentence described in Figure 3.1a. Figure 3.12 visualizes several key steps, where the rectangles denote nodes with particular types, among which the shaded ones are three competing hypotheses at each step. The solid lines and arrows indicate correct node and edge actions respectively, while the dashed indicate incorrect ones. For simplicity, only a small part of the search space is presented. At the very beginning, let us assume that the prefix “the

tire” has already been parsed, and “*the*_{/⊥} *tire*_{/⊥}” is one of the assignments in the buffer. In the next step, we predict possible nodes ending at the token “*maker*”. Figure 3.12b shows three possible candidates: “(*tire maker*)_{/ORG}”, “(*tire maker*)_{/PER}”, and “*maker*_{/⊥}”. They are then appended to their preceding partial configurations respectively as illustrated by the three lines.

As we continue, suppose we are at the token “1,400” (Figure 3.12c and 3.12d). At this point we can propose multiple nodes ending at this token with various lengths. Assuming “1,400_{/PER}”, “1,400_{/⊥}” and “(*employs 1,400*)_{/PER}” are three possible assignments, the algorithm then appends them to the partial assignments in the buffers of the tokens “*employs*” and “*still*”, respectively. The algorithm then links the newly identified nodes to the previous ones in the same configurations. In this example, the only preceding mention is “(*tire maker*)_{/ORG}”. Finally, “1,400_{/PER}” will be preferred by the model since there are more indicative context features for EMP-ORG relation between “(*tire maker*)_{/PER}” and “1,400_{/PER}”. By connecting the path of solid lines and arrows we obtain the final structure in Figure 3.1a.

3.6 Features

We develop two types of features, namely *local features* and *global features*. We first introduce the definition of local and global features in this chapter respectively, and then describe the implementation details later. Recall that in the linear model defined in Eq. 3.1, $\mathbf{f}(x, y)$ denotes the features extracted from the input instance x along with configuration y . In general, each feature instance f in \mathbf{f} is a function $f : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ that maps x and y to a feature value. Local features are only related to the predictions on individual triggers or arguments. In the case of unigram tagging for trigger labeling, each local feature takes the form of $f(x, i, y_{g(i)})$, where i denotes the index of the current token, and $y_{g(i)}$ is its trigger label. In practice, it is convenient to define the local feature function as an indicator function, for example:

$$f_{101}(x, i, y_{g(i)}) = \begin{cases} 1 & \text{if } y_{g(i)} = \mathbf{Attack} \text{ and } x_i = \text{“fire”} \\ 0 & \text{otherwise} \end{cases}$$

The global features, by contrast, involve longer range of the output structure. For example, each global feature function can take the form of $f(x, i, k, y)$, where i and k denote the indices of the last token of the current node, and the index of an argument candidate or a previous entity mention for edge classification, respectively. The following indicator function is a simple example of global features:

$$f_{201}(x, i, k, y) = \begin{cases} 1 & \text{if } y_{g(i)} = \mathbf{Attack} \text{ and} \\ & y \text{ has only one } \mathbf{Attacker} \\ 0 & \text{otherwise} \end{cases}$$

This feature function is triggered when the current token is classified as an **Attack** trigger, and it only has one **Attacker** argument so far.

3.6.1 Local Features

In general there are four kinds of local features: trigger features (token-based), event argument features, entity mention features (token-based and segment-based), and relation features. The input part of the features except token-based entity mention features are identical to those described in our baseline systems in Section 3.2.

3.6.1.1 Trigger Features

The local feature function for trigger labeling can be factorized as $f(x, i, y_{g(i)}) = p(x, i) \circ q(y_{g(i)})$, where $p(x, i)$ is a predicate about the input, which we call text feature, and $q(y_{g(i)})$ is a predicate on the trigger label. In practice, we define two versions of $q(y_{g(i)})$:

$$q_0(y_{g(i)}) = y_{g(i)} \text{ (event subtype)}$$

$$q_1(y_{g(i)}) = \text{event type of } y_{g(i)}$$

$q_1(y_{g(i)})$ is a backoff version of the standard unigram feature. Some text features for the same event type may share a certain distributional similarity regardless of the subtypes. For example, if the nearest entity mention is a “*Company*”, the current token is likely to be **Personnel** no matter whether it is **End-Position** or

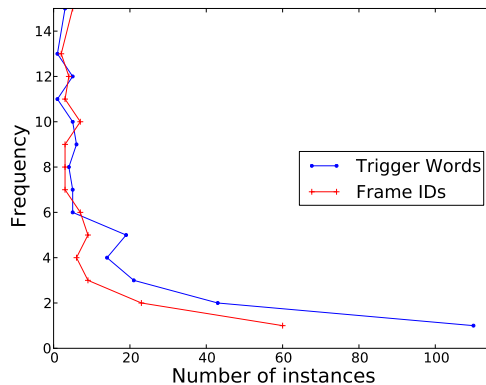


Figure 3.13: Distribution of triggers and their frames.

Start-Position.

One major challenge of constructing information networks is the data sparsity problem in extracting event triggers. For instance, in the sentence: “*Others were mutilated beyond recognition.*” The **Injure** trigger “*mutilated*” does not occur in our training data. But there are some similar words such as “*stab*” and “*smash*”. We utilize FrameNet [104] to solve this problem. FrameNet is a lexical resource for semantic frames. Each frame characterizes a basic type of semantic concept, and contains a number of words (lexical units) that evoke the frame. Many frames are highly related with ACE events. For example, the frame “*Cause_harm*” is closely related with **Injure** event and contains 68 lexical units such as “*stab*”, “*smash*” and “*mutilate*”. Figure 3.13 compares the distributions of trigger words and their frame identifiers in the training data. We can clearly see that the trigger word distribution suffers from the long-tail problem, while Frames reduce the number of triggers which occur only once in the training data from 100 to 60 and alleviate the sparsity problem. For each token, we exploit the frames that contain the combination of its lemma and POS tag as features. For the above example, “*Cause_harm*” will be a feature for “*mutilated*”. We only consider tokens that appear in at most 2 frames, and omit the frames that occur fewer than 20 times in our training data.

3.6.1.2 Argument Features

Similarly, the local feature function for argument labeling can be represented as $f(x, i, k, y_{g(i)}, y_{h(i,k)}) = p(x, i, k) \circ q(y_{g(i)}, y_{h(i,k)})$, where $y_{h(i,k)}$ denotes the argument assignment for the edge between trigger word i and argument candidate e_k . We define two versions of $q(y_{g(i)}, y_{h(i,k)})$:

$$q_0(y_{g(i)}, y_{h(i,k)}) = \begin{cases} y_{h(i,k)} & \text{if } y_{h(i,k)} \text{ is Place,} \\ & \text{Time or } \perp \\ y_{g(i)} \circ y_{h(i,k)} & \text{otherwise} \end{cases}$$

$$q_1(y_{g(i)}, y_{h(i,k)}) = \begin{cases} 1 & \text{if } y_{h(i,k)} \neq \perp \\ 0 & \text{otherwise} \end{cases}$$

It is notable that **Place** and **Time** arguments are applicable and behave similarly to all event subtypes. Therefore features for these arguments are not conjuncted with trigger labels. $q_1(y_{h(i,k)})$ can be considered as a backoff version of $q_0(y_{h(i,k)})$, which does not discriminate different argument roles but only focuses on argument identification.

3.6.1.3 Entity Mention Features

In the segment-based algorithm, we can use segment-based features to directly evaluate the properties of each entity mention instead of the individual tokens it contains. This is more natural way of modeling entity mentions than traditional token-based tagging such as [52]. The following is an example of segment-based feature:

$$f_{001}(x, y, i) = \begin{cases} 1 & \text{if } x_{[y.p_i, \hat{y}.q_i]} = \text{tire maker} \\ & y.t_{(i-1)}, y.t_i = \perp, \text{ORG} \\ 0 & \text{otherwise} \end{cases}$$

This feature is triggered if the labels of the $(i - 1)$ -th and the i -th segments are “ \perp, ORG ”, and the text of the i -th segment is “*tire maker*”.

1. **Gazetteer features**

Entity type of each segment based on matching a number of gazetteers including persons, countries, cities and organizations.

2. **Case features**

The capitalization information about the segment, one of *initial-capitalized*, *lower case*, or *mixture*.

3. **Contextual features**

The unigrams and bigrams of the text and part-of-speech tags in the window of size 2. Example: for “*tire maker*” in the example figure, $w_{(2)} = \textit{employs}$.

4. **Parsing-based features**

Features that are derived from constituent parse tree.

- Phrase label of the common ancestor of the entity mention’s tokens. Example: the phrase label of “*tire maker*” is NP.
- Depth of the common ancestor of the entity mention’s tokens. Example: the depth of “*tire maker*” is 1 in the parse tree.
- The entity mention exactly matches a base phrase or is a suffix of the phrase. Example: “*tire maker*” is a suffix of a NP phrase
- The head words of the entity mention and its neighbor phrases. Example: $\textit{head}(\textit{tire maker}) = \textit{maker}$, and $\textit{head}_1(\textit{tire maker}) = \textit{still}$.

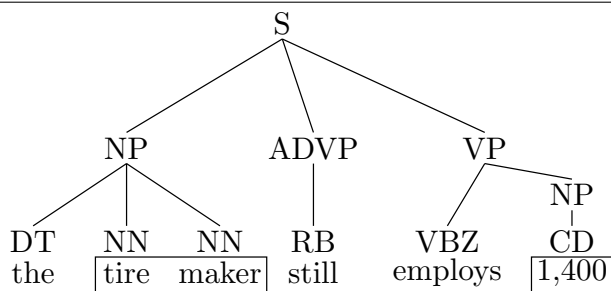


Figure 3.14: Segment-based local entity mention features.

Furthermore, we can still obtain token-based features by converting each entity triple $\langle p_i, q_i, t_i \rangle$ to BILOU tags for each token within it. For instance, the entity mention “ $(\textit{tire maker})_{\text{ORG}}$ ” has B-ORG, L-ORG tags for the two tokens within it. The token-based features then are expressed by combination of contextual features of each token and bigram of the current and previous token tags. The segment-based entity mention features with an example sentence “the tire maker still employs 1400” are described in Figure 3.14.

3.6.2 Global Features

As we mentioned earlier, in addition to local features, we are interested in exploiting various global features. In this task, local features are only related to the predictions on individual triggers or arguments. They are identical to the features that we can make use of in local classifiers of the pipelined approach. Global features, on the contrary, involve longer range of the output structure, and usually engage multiple event triggers (global trigger features), or argument edges (global argument features), entity mentions (global entity features), or relations (global relation features).

3.6.2.1 Global Trigger Features

This type of feature captures the dependencies between two triggers within the same sentence. Table 3.5 summarizes the global features about triggers that we developed in our study.

Table 3.5: Global trigger features.

ID	Feature Description
1.	bigram of trigger types occur in the same sentence or the same clause
2.	binary feature indicating whether synonyms in the same sentence have the same trigger label
3.	context and dependency paths between two triggers conjuncted with their types

For instance: feature (1) captures the co-occurrence of trigger types. This kind of feature is motivated by the fact that two event mentions in the same sentence tend to be semantically coherent. As an example, from Table 3.6 we can see that **Attack** event often co-occur with **Die** event in the same sentence, but rarely co-occur with **Start-Position** event. Feature (2) encourages synonyms or identical tokens to have the same label. Feature (3) exploits the lexical and syntactic relation between two triggers. A simple example is whether an **Attack** trigger and a **Die** trigger are linked by the dependency relation *conj_and*.

Table 3.6: Top five event subtypes that co-occur with Attack event. Event triggers are highlighted by underscores.

Event Type	Probability	Example
Attack	0.34	it would not commit <u>fighting</u> forces to the <u>war</u> .
Die	0.14	<u>destroyed</u> a command and control post and <u>killed</u> a number of soldiers.
Transport	0.08	The mob <u>killed</u> them with machetes and spears before <u>fleeing</u> the area.
Injure	0.04	Two Marines were <u>injured</u> in the close-quarters <u>fighting</u> .
Meet	0.02	Portugal hosted a last ditch pre- <u>war</u> <u>summit</u> in the Azores islands.

3.6.2.2 Global Argument Features

This type of feature is defined over multiple arguments for the same or different triggers. Table 3.5 summarizes the global features about arguments that we developed in our experiments.

Table 3.7: Global argument features.

ID	Feature Description
1.	context and dependency features about two argument candidates which share the same role within the same event mention
2.	features about one argument candidate which plays as arguments in two event mentions in the same sentence
3.	features about two arguments of an event mention which are overlapping
4.	the number of arguments with each role type of an event mention conjuncted with the event subtype
5.	the pairs of time arguments within an event mention conjuncted with the event subtype

Consider the following sentence:

Example 3.6.1. Trains running to southern Sudan were used to transport abducted women and children.

The **Transport** event mention “*transport*” has two **Artifact** arguments, namely “*women*” and “*children*”. The dependency edge *conj_and* between “*women*” and “*children*” indicates that they should play the same role in the event mention. The triangle structure in Figure 3.15a is an example of feature (1) for the above example.

This feature encourages entities that are linked by dependency relation *conj_and* to play the same role **Artifact** in any **Transport** event.

Similarly, Figure 3.15b depicts an example of feature (2) for the sentence:

Example 3.6.2. In Baghdad, a cameraman **died** when an American tank fired on the Palestine Hotel.

In this example, an entity mention is **Victim** argument to **Die** event and **Target** argument to **Attack** event, and the two event triggers are connected by the typed dependency *advcl*. Here *advcl* means that the word “fired” is an adverbial clause modifier of “died” [105].

Figure 3.15c shows an example of feature (3) for the following sentence:

Example 3.6.3. Barry Diller **resigned** as co-chief executive of Vivendi Universal Entertainment.

The job title “*co-chief executive of Vivendi Universal Entertainment*” overlaps with the **Organization** mention “*Vivendi Universal Entertainment*”. The feature in the triangle shape can be considered as a soft constraint such that if a *Job-Title* mention is a **Position** argument to an **End-Position** trigger, then the **Organization** mention that appears at the end of it should be labeled as an **Entity** argument for the same trigger.

Feature (4-5) are based on the statistics about different arguments for the same trigger. For instance, in many cases, a trigger can only have one **Place** argument. If a partial configuration mistakenly classifies more than one entity mention as **Place** arguments for the same trigger, then it will be penalized.

3.6.2.3 Global Entity Mention Features

The following features involve multiple entity mentions are extracted once a new segment is appended during the decoding.

Coreference consistency

Coreferential entity mentions should be assigned the same entity type. We determine high-recall coreference links between two segments in the same sentence using some simple heuristic rules:

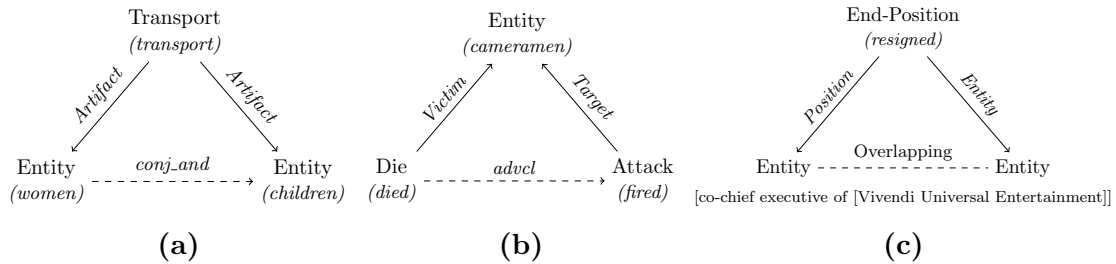


Figure 3.15: Illustration of the global features in Table 3.7 for the sentences “*Barry Diller resigned as co-chief executive of Vivendi Universal Entertainment*” and “*In Baghdad, a cameraman died when an American tank fired on the Palestine Hotel.*”

- String values of the two segments exactly or partially match. For example, “*Obama*” and “*Barack Obama*” have a partial match.
- A pronoun (e.g., “*their*”, “*it*”) refers to previous entity mentions. For example, in “*they have no insurance on their cars*”, “*they*” and “*their*” should have the same entity type.
- A relative pronoun (e.g., “*which*”, “*that*”, and “*who*”) refers to the noun phrase it modifies (if they are dominant by the same clause node in the parsing tree). For example, in “*the starting kicker is Nikita Kargalskiy, who may be 5,000 miles from his hometown in Russia*”, “*Nikita Kargalskiy*” and “*who*” should both be labeled as persons.

Then we encode a global feature to check whether two coreferential segments share the same entity type. This feature is particularly effective for pronouns because their contexts alone are often not informative.

Neighbor coherence

Neighboring entity mentions tend to have coherent entity types. For example, in “*Barbara Starr was reporting from the Pentagon*”, “*Barbara Starr*” and “*Pentagon*” are connected by a dependency link *prep_from* and thus they are unlikely to be a pair of PER mentions. We consider two senses of neighbor: (i) the first entity mention before the current segment, and (ii) two segments are connected by a single

word or a dependency link. We take the entity types of the two segments and the linkage as a global feature.

Part-of-whole consistency

If an entity mention is syntactically part of another mention (connected by a *prep_of* dependency link), they should be assigned the same entity type. For example, in “*some of Iraq’s exiles*”, “*some*” and “*exiles*” are both PER mentions; in “*one of the town’s two meat-packing plants*”, “*one*” and “*plants*” are both FAC mentions; in “*the rest of America*”, “*rest*” and “*America*” are both GPE mentions.

3.6.2.4 Global Relation Features

Relation edges can also share inter-dependencies or obey soft constraints. We extract the following relation-centric global features when a new relation hypothesis is made during decoding.

Role coherence

If an entity mention is involved in multiple relations with the same type, then its roles should be coherent. For example, a person mention is unlikely to have more than one employer. However, a GPE mention can serve as a physical location for multiple entity mentions. We combine the relation type and the entity mention’s argument roles as a global feature, as shown in Figure 3.16a.

Triangle constraint

Multiple entity mentions are unlikely to be fully connected with the same relation type. We use a negative feature to penalize any configuration that contains this type of structure. An example is shown in Figure 3.16b.

Inter-dependent compatibility

If two entity mentions are connected by a dependency link, they tend to have compatible relations with other entities. For example, in Figure 3.16c, the *conj_and* dependency link between “*Somalia*” and “*Kosovo*” indicates they may share the same relation type with the third entity mention “*forces*”.

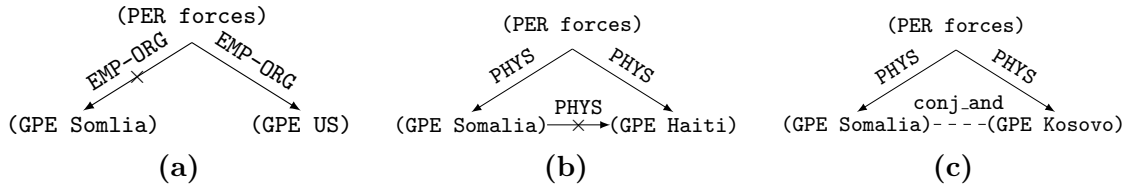


Figure 3.16: Examples of global relation features.

Table 3.8: Frequent overlapping relation and event types.

Freq.	Relation Type	Event Type	Arg-1	Arg-2	Example
159	Physical	Transport	Artifact	Destination	He _(arg-1) was escorted _(trigger) into Iraq _(arg-2) .
46	Physical	Attack	Target	Place	Many people _(arg-1) were in the cafe _(arg-2) during the blast _(trigger) .
42	Agent-Artifact	Attack	Attacker	Instrument	Terrorists _(arg-1) might use _(trigger) the devices _(arg-2) as weapons.
41	Physical	Transport	Artifact	Origin	The truck _(arg-1) was carrying _(trigger) Syrians fleeing the war in Iraq _(arg-2) .
33	Physical	Meet	Entity	Place	They _(arg-1) have reunited _(trigger) with their friends in Norfolk _(arg-2) .
32	Physical	Die	Victim	Place	Two Marines _(arg-1) were killed _(trigger) in the fighting in Kut _(arg-2) .
28	Physical	Attack	Attacker	Place	Protesters _(arg-1) have been clashing _(trigger) with police in Tehran _(arg-2) .
26	ORG-Affiliation	End-Position	Person	Entity	NBC _(arg-2) is terminating _(trigger) freelance reporter Peter Arnett _(arg-1) .

Neighbor coherence

Similar to the entity mention neighbor coherence feature, we also combine the types of two neighbor relations in the same sentence as a bigram feature.

3.6.2.5 Joint Relation-Event Feature

By extracting the three fundamental IE components in a joint search space, we can utilize joint features over multiple components in addition to factorized features in pipelined approaches. For instance, we can make use of joint features between relations and events, given the fact that relations are often ending or starting states of events [106]. Table 3.8 shows the most frequent overlapping relation and event types in our training data. In each partial structure y' during the search, if both

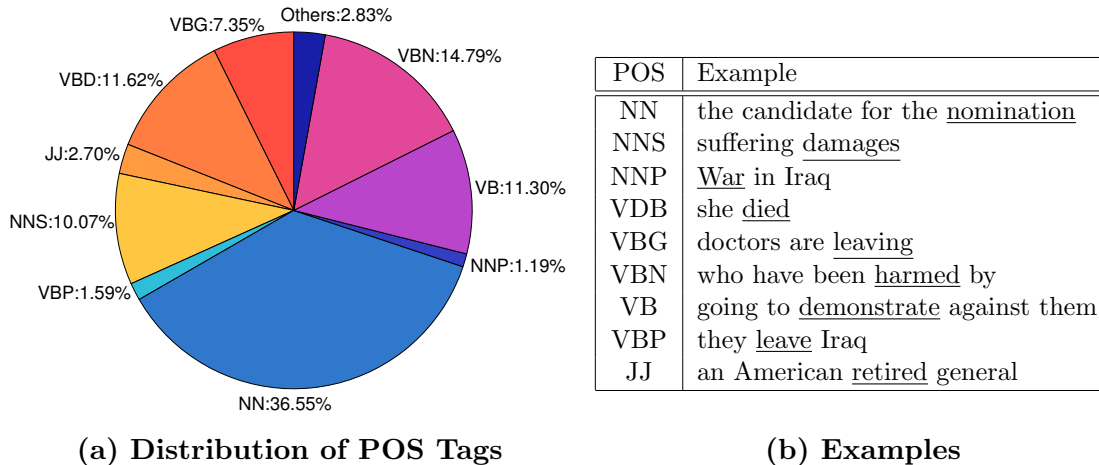
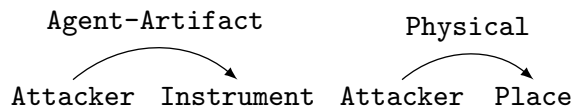


Figure 3.17: Distribution of event trigger w.r.t. POS tags.

arguments of a relation participate in the same event, we compose the corresponding argument roles and relation type as a joint feature for y' . For example, for the structure in Figure 3.2, we obtain the following joint relation-event features:



3.7 Experiments

Our evaluation consists of three parts: In sub-section 3.7.2 we first evaluate the performance of the token-based decoder under the setting that argument candidates are given. In sub-section 3.7.3 we extract entity mentions and relations by using the segment-based decoder to create an end-to-end relation extraction system, and conduct experiments on both ACE'04 and ACE'05 corpus. Finally, in sub-section 3.7.4 we add event triggers and arguments to the framework to construct a complete joint model, and conduct experiments only on ACE'05 corpus, since ACE'04 corpus does not contain any event annotations.

3.7.1 Evaluate Setup

We first conduct experiments to evaluate the joint framework with the token-based decoder, where the argument candidates are given by either gold-standard annotation or system prediction. For this purpose, we use the ACE'05 corpus as our

test-bed. And for a fair comparison, we used the same test set with 40 newswire articles (672 sentences, 440 events, 703 arguments) as in [6, 7] for the experiments, and randomly selected 30 other documents (863 sentences, 505 events, 806 arguments) from different genres as the development set. The rest 529 documents (14.8k sentences, 4.4k events, 6.6k arguments) are used for training. It is worth noting that the trigger words of events can be of many different part-of-speech tags besides verb. Figure 3.17 illustrates the distribution of event triggers with respect to part-of-speech tags. In fact, the majority of triggers are noun (62.6%) and verb (46.6%), but there exist some exceptions such as adjective (JJ), pronoun (PRP), and adverb (RB). JJ accounts for 2.7% of triggers, and others only account for 2.8%. Following previous work [6–8], we use the following criteria to determine the correctness of an event mention in system output.

- A trigger is correct if its event subtype and offsets match those of a reference trigger.
- An argument is correctly *identified* if its event subtype and offsets match those of any of the reference argument mentions.
- An argument is correctly *identified* and *classified* if its event subtype, offsets and argument role match those of any of the reference argument mentions.

Finally we use Precision (P), Recall (R) and F-measure (F_1) to evaluate the overall performance.

Then, we evaluate the joint framework with the segment-based decoder that jointly extract entity mentions, relations and events. Most previous work on ACE relation extraction has reported results on ACE’04 data set. ACE’05 made significant improvement on both relation type definition and annotation quality. Therefore we present the overall performance of our approaches on ACE’05 data. We removed two small subsets in informal genres *cts* (conversation telephone speech) and *un* (Usenet web forum), and then randomly split the remaining 511 documents into 3 parts: 351 for training, 80 for development, and the rest 80 for blind test. In order to compare with state-of-the-art systems we also performed the same 5-fold cross-validation on *bnews* and *nwire* subsets of ACE’04 corpus as in previous work. The statistics of these data sets are summarized in Table 3.9. We ran the Stanford

CoreNLP toolkit⁶ to automatically recover the true cases for lowercased documents. We use the standard F_1 measure to evaluate the performance of entity mention ex-

Table 3.9: Statistics about data set.

Data Set		# sentences	# mentions	# relations	# triggers	# arguments
ACE'05	Train	7,2k	26.4k	4,8k	2.8k	4.5k
	Dev	1,7k	6.3k	1.2k	0.7k	1.1k
	Test	1.5k	5.4k	1.1k	0.6k	1.0k
ACE'04		6.8k	22.7k	4.4k	N/A	N/A

traction and relation extraction. An entity mention is considered correct if its entity type is correct and the offsets of its mention head are correct. A relation mention is considered correct if its relation type is correct, and the head offsets of two entity mention arguments are both correct. As in [10], we excluded the DISC relation type, and removed relations in the system output which are implicitly correct via coreference links in order to conduct a fair comparison. Furthermore, we combine these two criteria to evaluate the performance of end-to-end entity mention and relation extraction.

3.7.2 Results of Token-based Decoding

We use the *harmonic mean* of the trigger's F_1 measure and argument's F_1 measure to evaluate the performance on the development set. Figure 3.18 shows the training curves of the averaged perceptron with respect to the performance on the development set when the beam size is 8. As we can see both curves converge around iteration 20 and the global features improve the overall performance, compared to its counterpart with only local features. Therefore we set the number of iterations as 20 in the remaining experiments.

The beam size is an important hyper parameter in both training and test. Larger beam size will increase the computational cost while smaller beam size may reduce the performance. Table 3.10 shows the performance on the development set with several different beam sizes. When beam size = 8, the algorithm achieved the highest performance on the development set with trigger $F_1 = 68.7$, argument $F_1 = 51.8$, and harmonic mean = 59.1. When the size is increased to 32, the accuracy

⁶<http://nlp.stanford.edu/software/corenlp.shtml> (Date Last Accessed, March, 10, 2015)

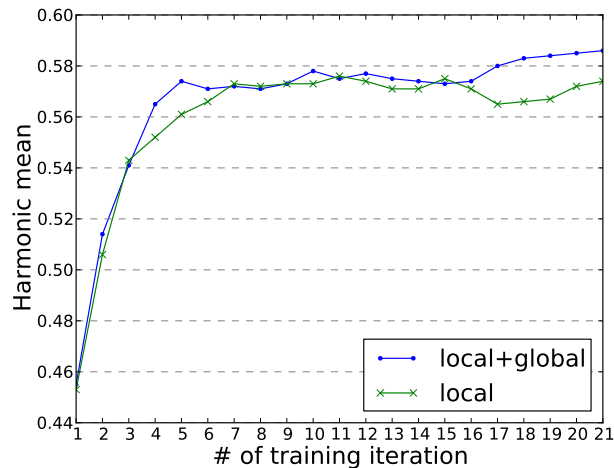


Figure 3.18: Training curves on development set.

Table 3.10: Comparison of training time and accuracy on dev set.

Beam size	1	2	4	8	16	32
Training time (sec)	579	1,228	2,509	5,040	10,279	20,215
Harmonic mean	57.5	57.8	58.6	59.1	57.8	58.6

was not improved. There may be two reasons: 1) The overfilling problem. When we have larger beam, the search capacity during the training is larger. But this cannot directly effect the performance on a blind test data. 2) Since the whole task is quite challenging and even the annotation quality is limited, many easy instances can already be handled when the beam size is relatively small. But increasing the beam size does not improve the chance of solving difficult instances.

Based on this observation, we chose beam size as 8 for the rest experiments. Table 3.11 shows the overall performance on the blind test set. Our pipelined baseline outperforms the sentence-level system reported in previous work [6–8]. Among those [8] used the ground-truth entity information in the ACE corpus. The joint framework with local features outperforms the pipelined baseline especially on arguments, and adding global features further significantly improved the overall performance.

In addition to the standard data splitting, we also tested our methods in the setting of 5-fold cross-validation. For this experiment, we only chose news (nw) and broadcast news (bn) subsets so as to rule out domain shifts caused by other informal

Table 3.11: Overall performance with gold-standard argument candidates (entity mention, ACE value, and timex).

Methods	Trigger Identification (%)			Trigger Identification + classification (%)			Argument Identification (%)			Argument Role (%)		
	P	R	F ₁	P	R	F ₁	P	R	F ₁	P	R	F ₁
Sentence-level in [8]	N/A			67.6	53.5	59.7	46.5	37.15	41.3	41.0	32.8	36.5
Pipelined Baseline	76.2	60.5	67.4	74.5	59.1	65.9	74.3	37.9	50.2	65.1	33.2	44.0
Joint w/ local	80.7	61.8	70.0	77.2	59.1	66.9	73.2	42.4	53.7	67.4	39.0	49.4
Joint w/ local + global	79.7	62.5	70.1	75.9	59.5	66.8	73.4	46.5	57.0	67.9	43.0	52.7

Table 3.12: Overall performance from 5-fold cross-validation.

Methods	Trigger Identification (%)			Trigger Identification + classification (%)			Argument Identification (%)			Argument Role (%)		
	P	R	F ₁	P	R	F ₁	P	R	F ₁	P	R	F ₁
Pipelined Baseline	77.7	61.9	68.9	75.5	60.2	66.9	76.9	37.3	50.2	68.5	33.2	44.8
Joint w/ local	77.7	66.3	71.6	74.8	63.8	68.9	74.8	46.2	57.1	69.4	42.9	53.0
Joint w/ local + global	77.3	67.0	71.8	74.2	64.3	68.9	73.5	48.5	58.4	69.0	45.5	54.9

subsets. Table 3.12 summarizes the 5-fold cross-validation results. From this table we can observe that the performance of the three methods have the same trend in the previous setting. However, the overall accuracy is higher. Specifically, the F₁ score of the final event argument extraction is 2.1% higher. This is because in the standard setting 1) the training data is a mixture of different genres, and 2) the actual test data is relatively small.

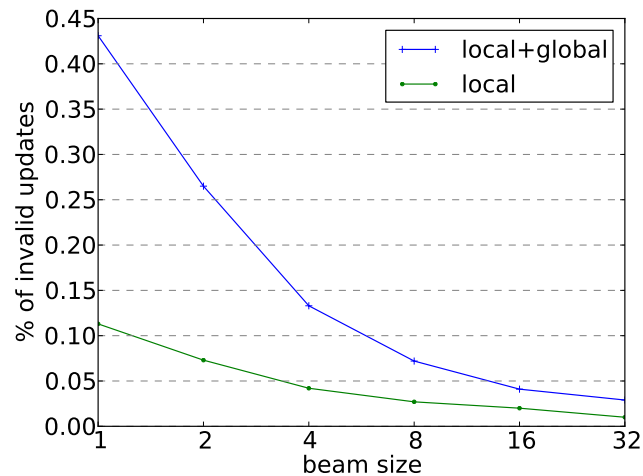
**Figure 3.19: Percentage of “invalid updates” [101] in standard perceptron.**

Table 3.13: Comparison between the performance (%) of standard-update and early-update with global features. k stands for beam size.

Strategy	F ₁ on Dev		F ₁ on Test	
	Trigger	Argument	Trigger	Argument
Standard ($k = 1$)	68.3	47.4	64.4	49.8
Early ($k = 1$)	68.9	49.5	65.2	52.1
Standard ($k = 4$)	68.4	50.5	67.1	51.4
Early ($k = 4$)	67.9	51.5	67.5	52.7

Why Early Update

“Invalid update” [101] is the parameter update when the model score of the gold-standard is higher than the score of the top-ranked hypothesis. It can reinforce search error rather than fixing violation. It strongly (anti-)correlates with search quality and learning quality. Figure 3.19 depicts the percentage of invalid updates in standard-update with and without global features, respectively. With global features, there are numerous invalid updates when the beam size is small. The ratio decreases monotonically as beam size increases. The model with only local features made much smaller numbers of invalid updates, which suggests that the use of global features makes the search problem much harder. This observation justifies the application of early-update in this work. To further investigate the difference between early-update and standard-update, we tested the performance of both strategies, which is summarized in Table 3.13. As we can see the performance of standard-update is generally worse than early-update. When the beam size is increased ($k = 4$), the gap becomes smaller as the ratio of invalid updates is reduced.

3.7.3 Results of End-to-End Relation Extraction

In this experiment, we develop an end-to-end relation extraction system with the joint framework using the segment-based decoder. As we already mentioned, generally larger beam size can yield better performance but increase training and decoding time. As a tradeoff, we set the beam size as 8 throughout the experiments. Figure 3.20 shows the learning curves on the development set, and compares the performance with and without global features. From these figures we can clearly see

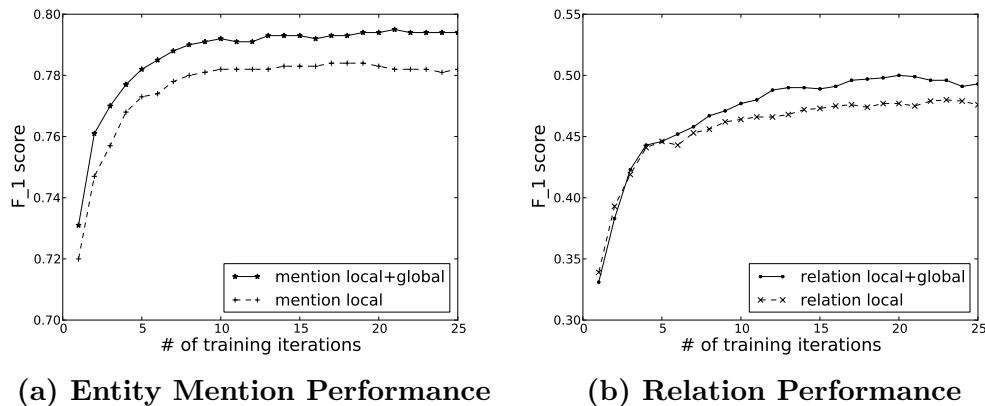


Figure 3.20: Learning curves on development set.

that global features consistently improve the extraction performance of both tasks. We set the number of training iterations as 22 based on these curves.

Table 3.14: Overall performance on ACE’05 corpus.

Model	Entity Mention (%)			Relation (%)			Entity Mention + Relation (%)		
	P	R	F ₁	P	R	F ₁	P	R	F ₁
Pipelined	83.2	73.6	78.1	67.5	39.4	49.8	65.1	38.1	48.0
Joint w/ Local	84.5	76.0	80.0	68.4	40.1	50.6	65.3	38.3	48.3
Joint w/ Global	85.2	76.9	80.8	68.9	41.9	52.1	65.4	39.8	49.5
Annotator 1	91.8	89.9	90.9	71.9	69.0	70.4	69.5	66.7	68.1
Annotator 2	88.7	88.3	88.5	65.2	63.6	64.4	61.8	60.2	61.0
Inter-Agreement	85.8	87.3	86.5	55.4	54.7	55.0	52.3	51.6	51.9

Table 3.14 shows the overall performance of various methods on the ACE’05 test data. We compare our proposed method (Joint w/ Global) with the pipelined system (Pipelined), the joint model with only local features (Joint w/ Local), and two human annotators who annotated 73 documents in ACE’05 corpus. We can see that our approach significantly outperforms the pipelined approach for both tasks. The human F₁ score on end-to-end relation extraction is only about 70%, which indicates it is a very challenging task. Furthermore, the F₁ score of the inter-annotator agreement is 51.9%, only 2.4% above that of our proposed method.

Table 3.15 compares the performance on ACE’04 corpus. For entity mention extraction, our joint model achieved 79.7% on 5-fold cross-validation, which is comparable with the best F₁ score 79.2% reported by [52] on single-fold. However, [52] used some gazetteers and the output of other IE models as additional features. Ac-

Table 3.15: 5-fold cross-validation on ACE’04 corpus. Bolded scores indicate highly statistical significant improvement as measured by paired t-test ($p < 0.01$)

Methods	Entity Mention (%)			Relation (%)			Entity Mention + Relation (%)		
	P	R	F ₁	P	R	F ₁	P	R	F ₁
Chan and Roth 2011 [10]		-		42.9	38.9	40.8		-	
Pipelined Approach	81.5	74.1	77.6	62.5	36.4	46.0	58.4	33.9	42.9
Joint w/ Local	82.7	75.2	78.8	64.2	37.0	46.9	60.3	34.8	44.1
Joint w/ Global	83.5	76.2	79.7	64.7	38.5	48.3	60.8	36.1	45.3

According to [29], these external IE models included name taggers trained from other data sets such as Message Understanding Conference (MUC) corpus, and provided significant gains (about 1.3%). Since these gazetteers, additional data sets and external IE models are all not publicly available, it is not fair to directly compare our joint model with their results. For end-to-end entity mention and relation extraction, both the joint approach and the pipelined baseline outperform the best results reported by [10] under the same setting.

3.7.4 Results of Complete Joint Model

We finally combine all of the three subtasks into a complete joint model. In this experiment, in addition to the perceptron update method, we employ the following three loss functions in k-best MIRA method (see Section 3.3.1):

- The first one is F_1 loss. Given the gold-standard y and prediction z , it is calculated based on overall F_1 measure for a prediction:

$$L_1(y, z) = 1 - \frac{2 \cdot |y \cap z|}{|y| + |z|}$$

When counting the numbers, we treat each node or edge as a single unit. For example, in Figure 3.2, $|y| = 6$.

- The second one is 0-1 loss:

$$L_2(y, z) = \begin{cases} 1 & y \neq z \\ 0 & y = z \end{cases}$$

It does not discriminate the extent to which z deviates from y .

- The third loss function counts the difference between y and z :

$$L_3(y, z) = |y| + |z| - 2 \cdot |y \cap z|$$

Similar to F_1 loss function, it penalizes both missing and false-positive units. The difference is that it is sensitive to the size of y and z .

Based on the results of our development set, we trained all models with 21 iterations and chose the beam size to be 8. For the k-best MIRA updates, we set k as 3. Table 3.16 compares the overall performance of our approaches and baseline methods. The joint model with perceptron update outperforms the pipelined approach in Section 3.2, and further improves the joint event extraction system in [23] ($p < 0.05$ for entity mention extraction, and $p < 0.01$ for other subtasks, according to Wilcoxon Signed RankTest). For the k-best MIRA update, the L_3 loss function achieved better performance than F_1 loss and 0-1 loss on all sub-tasks except event argument extraction. It also significantly outperforms perceptron update on relation extraction and event argument extraction ($p < 0.01$). It is particularly encouraging to see the end output of an IE system (event arguments) has made significant progress (12.2% absolute gain over traditional pipelined approach).

The complete joint model searches for the best configuration through a very large search space. Recall that the worst-case time complexity of this model is $O(\hat{d} \cdot k \cdot s^2)$ (s stands for the number of tokens in the sentence), while the pipelined approach only takes $O(s)$ for extracting information nodes, and the number of edges

Table 3.16: Overall performance (%) on test set.

Methods	Entity Mention			Relation			Event Trigger			Event Argument		
	P	R	F ₁	P	R	F ₁	P	R	F ₁	P	R	F ₁
Pipelined Baseline	83.6	75.7	79.5	68.5	41.4	51.6	71.2	58.7	64.4	64.8	24.6	35.7
Pipeline + Li & et al.[23]				N/A			74.5	56.9	64.5	67.5	31.6	43.1
Li & Ji [24]	85.2	76.9	80.8	68.9	41.9	52.1	N/A					
Joint w/ Avg. Perceptron	85.1	77.3	81.0	70.5	41.2	52.0	67.9	62.8	65.3	64.7	35.3	45.6
Joint w/ MIRA w/ F ₁ Loss	83.1	75.3	79.0	65.5	39.4	49.2	59.6	63.5	61.5	60.6	38.9	47.4
Joint w/ MIRA w/ 0-1 Loss	84.2	76.1	80.0	65.4	41.8	51.0	65.6	61.0	63.2	60.5	39.6	47.9
Joint w/ MIRA w/ L ₃ Loss	85.3	76.5	80.7	70.8	42.1	52.8	70.3	60.9	65.2	66.4	36.1	46.8

to be classified in a pipelined approach is usually less than s since the number of entity mentions and triggers are far less than the number of tokens (see Table 3.9). To compare the running time of those two methods empirically, the following table compares the running of the pipelined baseline and our joint model on the test data:

System	Pipelined Baseline	Joint Model ($b = 1$)	Joint Model ($b = 8$)
Decoding Time (sec.)	32.5	142.8	800.1

This information is obtained by running both systems in the same computer. The numbers excluded the time consumption of pre-processing such as part-of-speech tagging, parsing and feature extraction. As we can see, the pure decoding time for the joint model is significantly more than the pipelined approach, as the search space of the latter is much larger. We leave improving the efficiency of the implementation to future work.

Feature Study

Table 3.17: Top features about event triggers.

Rank	Feature		Weight
1	Frame=Killing	Die	0.80
2	Frame=Travel	Transport	0.61
3	Physical(Artifact, Destination)		0.60
4	w_1 ="home"	Transport	0.59
5	Frame=Arriving	Transport	0.54
6	ORG-AFF(Person, Entity)		0.48
7	Lemma=charge	Charge-Indict	0.45
8	Lemma=birth	Be-Born	0.44
9	Physical(Artifact, Origin)		0.44
10	Frame=Cause_harm	Injure	0.43

Table 3.17 lists the most significant features about event triggers ranked by their weights. The 3rd, 6th, and 9th rows are joint relation-event features. For instance, *Physical(Artifact, Destination)* means the arguments of a **Physical** relation participate in a **Transport** event as **Artifact** and **Destination**. We can see that both the joint relation-event features and FrameNet based features are of vital importance to event trigger labeling. We tested the impact of each type of features by excluding them in the experiments of "MIRA w/ L_3 loss". We found that FrameNet

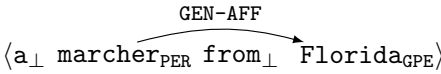
steps	hypotheses	rank
(a)	$\langle a_{\perp} \text{ marcher}_{\perp} \rangle$	1
	$\langle a_{\perp} \text{ marcher}_{\text{PER}} \rangle$	2
(b)	$\langle a_{\perp} \text{ marcher}_{\perp} \text{ from}_{\perp} \rangle$	1
	$\langle a_{\perp} \text{ marcher}_{\text{PER}} \text{ from}_{\perp} \rangle$	4
(c)	$\langle a_{\perp} \text{ marcher}_{\text{PER}} \text{ from}_{\perp} \text{ Florida}_{\text{GPE}} \rangle$	1
	$\langle a_{\perp} \text{ marcher}_{\perp} \text{ from}_{\perp} \text{ Florida}_{\text{GPE}} \rangle$	2
(d)	$\langle a_{\perp} \text{ marcher}_{\text{PER}} \text{ from}_{\perp} \text{ Florida}_{\text{GPE}} \rangle$ 	1
	$\langle a_{\perp} \text{ marcher}_{\perp} \text{ from}_{\perp} \text{ Florida}_{\text{GPE}} \rangle$	4

Figure 3.21: Two competing hypotheses for “*a marcher from Florida*” during joint extraction with global features.

based features provided 0.8% and 2.2% F_1 gains for event trigger and argument labeling respectively. Joint relation-event features also provided 0.6% F_1 gain for relation extraction.

3.7.5 Real Example

As a real example, for the partial sentence “*a marcher from Florida*” from the test data, the pipelined approach failed to identify “*marcher*” as a PER mention, and thus missed the GEN-AFF relation between “*marcher*” and “*Florida*”. Our joint model correctly identified the entity mentions and their relation. Figure 3.21 shows the details when the joint model is applied to this sentence. At the token “*marcher*”, the top hypothesis in the beam is “ $\langle \perp, \perp \rangle$ ”, while the correct one is ranked the second best. After the decoder processes the token “*Florida*”, the correct hypothesis is promoted to the top in the beam by the *Neighbor Coherence* features for PER-GPE pair. Furthermore, after linking the two mentions by GEN-AFF relation, the ranking of the incorrect hypothesis “ $\langle \perp, \perp \rangle$ ” is dropped to the 4-th place in the beam, resulting in a large margin from the correct one.

3.8 Analysis of Human Performance

The official ACE'05 training data is produced based on the efforts of multiple human annotators with multiple passes. To analyze the common errors of human annotation, we check the results of the first passes of two human annotators that prepared the ACE corpus, which includes 73 documents in total.

Entity Mention Annotation Errors

The most significant type of annotation error in entity mentions is the mention boundary error (281 instances). Many entity mentions are combinations of several smaller mentions. As a result, the mention structures can be very complicated, and it is even difficult for human being to make correct judgment on their boundaries. Some examples of this type of error are as follows:

- “*a waiting shed at the Davao City international airport_{FAC}*”. The annotators mistakenly considered the FAC mention as two separate mentions: “Davao City_{GPE}” and “airport_{FAC}”.
- “*according to documents filed in Los Angeles Superior Court_{ORG}*”. The annotators considered “*Los Angeles*” as a GPE mention, and tagged “*Court*” as an ORG mention.
- “*a Harvard Medical School_{ORG} graduate*”. The annotators considered “*Harvard*” as an ORG mention, and tagged “*School*” as another ORG mention.

The second remarkable type of error is the confusion between ORG and GPE (73 instances). This happens frequently when the mentions refer to the government of a country. For instance, in the following two sentences:

- “*other top Kremlin_{GPE} officials*”. “*Tahrir Square*” is mistakenly tagged as a LOC mention.
- “*two senior White House_{GPE} officials said*”. “*park*” is mistakenly tagged as a LOC mention.

Both GPE mentions were tagged as ORG by the annotators. There is also confusion between FAC and LOC (8 instances). According to the annotation guideline, FAC

mentions are primarily man-made structures designed for human habitation, storage, and transportation infrastructure and so on. On the other hand, LOC mentions are geographical or astronomical concepts such as river, mountain and so on. The name of Location is quite misleading since a building or a street is usually considered as “locations” in our daily life. For example:

- “*rallied on the city’s central Tahrir Square_{FAC}*”. “*Tahrir Square*” is mistakenly tagged as a LOC mention.
- “*show up at the park_{FAC} at Osaka Castle in the city’s downtown*”. “*park*” is mistakenly tagged as a LOC mention.

There are also noticeable confusion between ORG and PER (69 instances) when the entity mention refers to a group of people. For instance, “*cabinet*” refers to a group of high-ranking members of a government. And “*forces*” refers to an organized body of military personnel. They are often confused by the annotators. Moreover, the annotations of those mentions are even inconsistent in the gold standard.

Finally, some entity mentions are frequently neglected by the annotators, such as “*White House*_{GPE}”, “*elsewhere*_{GPE}”, “*elsewhere*_{LOC}”, and “*convoy*_{VEH}”.

Relation Annotation Errors

The most significant type of relation annotation error is the confusion between ORG-AFF and GEN-AFF (41 instances). ORG-AFF cares about the employment or affiliation relation between a PER mention and an ORG or GPE mention. GEN-AFF cares about a person’s citizenship, residence, and religion. For instance, according to the gold standard, the relation between “*US*” and “*troops*” in “*US troops*” should be ORG-AFF, since the employer of “*troops*” is “*US*”. On the other hand, the relation between “*Iraqi*” and “*scientist*” in “*an Iraqi chemical scientist*” should be GEN-AFF, since the citizenship of “*scientist*” is “*Iraq*”. The annotators often confuse about the above two cases. Another type of common error is the confusion between PHYS and PART-WHOLE (16 instances). For instance:

- In “*The three military checkpoints on the highway*”, the relation between “*checkpoints*” and “*highway*” is PART-WHOLE. But the annotators considered

it to be PHYS.

- In “*the area around the Tigris and Euphrates rivers*”, the relation between “*rivers*” and “*area*” is PHYS. But the annotators considered it to be PART-WHOLE.

Event Annotation Errors

Some event triggers are pronouns such as “this”, and “it”. It requires an annotator or an IE system to conduct event coreference. We found that the annotators usefully perform poorly on those cases. For example, in the following sentence:

- “*Nobody questions whether this_{Attack} is right or not*”.

The annotators failed to identify “*this*” as an **Attack** trigger since it requires extra effort to link it to an **Attack** event in the previous context. In addition, there exists confusion between **Die** and **Attack**. For instance, in the following sentences:

- “*The mob dragged out three members of a family and killed_{Attack} them*”.
- “*alleged plot to assassinate_{Attack} Megawati Soekarnoputri in 1999*”.

The annotators mistakenly considered the **Attack** triggers as **Die** triggers (9 instances), since the result of those events would be death.

For argument labeling, the annotators often made mistakes when the distance between the argument and the trigger is long, or the argument link is implicitly expressed. Those cases are also difficult for a learning based algorithm. For example, in the following sentences,

- “*US President George W. Bush condemned the attack on innocents in Israel. White House spokesman Ari Fleischer said, adding that his message to the terrorists is: Their efforts will not be successful.*”.
- “*the convoy was caught in a crossfire and three diplomats were hurt*”.

The arguments are syntactically far from the triggers. As a result, the annotators failed to connect them by argument links. In another sentence: “*Hamas issued a chilling warning against those taking part in the war against Iraq*”, Even the distance between “war” and “Iraq” is short, it does not explicitly entail that “Iraq” is a **Place** argument of “war”. Therefore it requires inference with background knowledge.

3.9 Analysis of Remaining Challenges

Although our system significantly improved the performance of end-to-end extraction, it is far from perfect. In this section, we make analysis about the remaining errors, and summarize them into several categories. Figure 3.22 summarizes the percentage of each category, which is calculated based on 200 random examples from the experiment results.

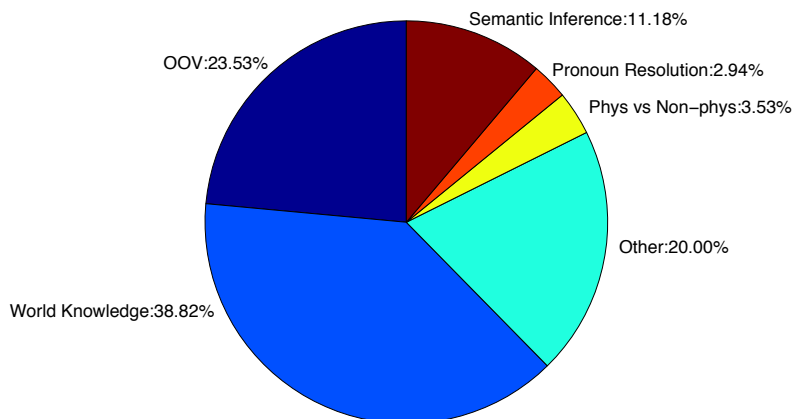


Figure 3.22: Distribution of different types of challenges.

Handle Out-of-Vocabulary and Sparsity Issue

OOV (shorthand for Out-of-Vocabulary) problem is a common issue in natural language processing. The task of information extraction also suffers from this problem. In fact, the recall of our system and baseline systems is significantly lower than the precision. In our test data, there are 654 event triggers in total, among which 77 (11.7%) trigger words and their lemmas do not appear in the training data.

Moreover, 140 (21.5%) trigger words in the test data appear fewer than twice in the training data. Some examples of OOV trigger words are

- people have been rioting_{Attack} in benton harbor.
- The airdrop_{Transport} - one of the biggest paratroop drops in decades.
- he embezzled_{Transfer-Money} hundreds of millions of dollars from aid funds.
- Marines found the mutilated_{Injure} body.

- Taken the risk of supporting a Shiite insurrection_{Demonstrate} at that point.
- Now Willie Williams the girl’s father is qharthd_{Charge-Indict} attempted murder.

Capture World Knowledge

Word knowledge is necessary for improving the performance of IE to a completely new level. In many cases, when annotators performing annotation, their world knowledge plays a key role in making judgment. For example, some words act as triggers for a certain types of events only when they appear together with some particular arguments:

- “*Williams picked up the child again and this time, threw_{Attack} her out the window.*” The word “*threw*” is used as an **Attack** event trigger because the **Victim** argument is a “*child*”.
- “*Ellison to spend \$10.3 billion to get_{Merge.Org} his company.*” The common word “*get*” is tagged as a trigger of *Merge_Org*, because its object is “*company*”.
- “*We believe that the likelihood of them using_{Attack} those weapons goes up.*” The word “*using*” can be considered an **Attack** event trigger because the **Instrument** argument is “*weapons*”.

Distinguish Physical and Non-physical Events

Some event trigger words are used by narrators to express logical or emotional events rather than physical events. It is difficult but important to distinguish them. For example, in the sentence:

- “*we are paying great attention to their ability to defend_{Attack} on the ground.*”,

our system failed to extract “*defend*” as an **Attack** trigger. In the training data, “*defend*” appears multiple times, but none of them is tagged as **Attack**. For instance, in the sentence:

- “*North Korea could do everything to defend itself.*”

“*defend*” is not an **Attack** trigger since it does not relate to physical actions in a war. Conversely, In another example:

- “*it is still hurts me to read this.*”

Our system mistakenly tagged “hurts” as an **Attack** trigger, but it only indicates negative emotion. This challenge calls for deeper understanding of the contexts.

Perform Pronoun Resolution

Pronouns are used to refer to actual events. Event coreference is necessary to recognize them correctly. For example, in the following two sentences from the same document:

- “*It’s important that people all over the world know that we don’t believe in the war/**Attack**.*”
- “*Nobody questions whether this/**Attack** is right or not.*”

“*this*” refers to “*war*” in its preceding contexts. Without accurate pronoun resolution, it is difficult to recognize it as an **Attack** event trigger.

Perform Semantic Inference

In some cases, identifying event arguments requires sophisticated semantic inference when there does not exist direct lexical or syntactic evidence to support the argument edges. For example, in the following sentence:

- “*Allied successes were marred by the collision of two Royal Navy helicopters over the Persian Gulf in which all six British crew members and one American were killed.*”

It is trivial to recognize “*killed*” as a trigger of **Die**. However, recognizing “*helicopters*” and “*Persian Gulf*” as its **Instrument** argument and **Place** argument is difficult. The lexical and syntactic structure of this sentence suggests that it is more easier to classify “*helicopters*” and “*Persian Gulf*” as arguments of “*collision*”, and propagate them to “*killed*”. However, since “*collision*” is not an event defined in ACE, even using global features is insufficient. Semantic inference is needed to recognize the casual relation between “*collision*” and “*killed*”, and identify “*helicopters*” and “*Persian Gulf*” as arguments to “*collision*”, and finally propagate them to “*killed*”. In another example:

- Negotiations between Washington and Pyongyang on their nuclear dispute have been set for April 23 in Beijing and are widely seen here as a blow to Moscow efforts to stamp authority on the region by organizing such a meeting.

Our system correctly extracted “*meeting*” as a trigger of **Meet**, but failed to identify “*Washington*” and “*Pyongyang*” as its **Entity** arguments. If we known “*Negotiations*” is a sub-event of “*meeting*” and “*Washington*” and “*Pyongyang*” are arguments of “*Negotiations*”, it would be more easier to infer that they are also arguments of “*meeting*” event.

3.10 Discussion

Beginning by an overview of the baseline IE pipeline, in this chapter we introduced the joint extraction framework based on a novel formulation of *information networks*. We described the training method of this framework and introduced two types of decoding algorithms: token-based decoding and segment-based decoding. The token-based decoding is conceptually and computationally simpler than the segment-based one. Although it can yield better performance than the traditional pipelined approach, It can only be applied in the setting where entity mentions are given because of the problem of synchronization. By contrast, in each step of the segment-based algorithm, segments with various lengths ending the current token are proposed, so that the search can be synchronized by the token indices. Therefore, we are able to build a system that can jointly extract entity mentions, relations and events. Within this framework, we exploited various global features to capture the dependencies over multiple local predictions. The experiments on ACE’05 and ACE’04 corpora showed the advantages of this new framework. Our final model achieved state-of-the-art performance at each stage of the extraction, and outperforms the system using token-based decoder. To the best of our knowledge, this is the first work that extracts the three subtasks in a joint model.

CHAPTER 4

Joint Inference for Cross-document Information Extraction

In this chapter we exploit cross-document dependencies to improve information extraction from sentence-level extractors. We describe a simple yet effective approach to conduct global inference with an Integer Linear Programming (ILP) formulation. Without using any additional labeled data, this new method obtained 13.7%-24.4% user browsing cost reduction over a sentence-level IE system.

4.1 Cross-document Information Extraction

First, we define the task of cross-document IE by extending the ACE terminology from single document to cross-document setting as follows: given a collection of source documents, a cross-document IE system should produce a knowledge base of unique facts. We apply a sentence-level English ACE single-document IE system [7] as our baseline to extract facts from individual documents. Finally we combine entity mentions using co-reference chain and string matching to create a cross-document information network, where each link is a specific occurrence of relation or event predicate between two entity nodes with a local confidence value.

In the remaining part of this section, we will discuss the dependencies and constraints of the IE output in detail, and then present the ILP-based global reasoning approach to enhance IE performance.

Global Dependency Constraints

We explore constraints across various types of relations and events. Let L^i denote a unique relation or event predicate linking two entities A and B . We consider three types of dependency as depicted in Figure 4.1.

We compute point-wise mutual information (PMI) to automatically estimate the pairwise dependency between any two types of links from ACE'05 training data:

Portions of this chapter previously appeared as: Q. Li, S. Anzaroot, W.-P. Lin, X. Li, and H. Ji, "Joint inference for cross-document information extraction," in *Proc. Int. Conf. on Inform. and Knowledge Manage.*, Glasgow, UK, 2011, pp. 2225–2228.

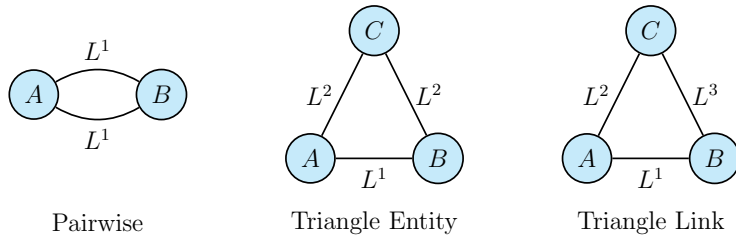


Figure 4.1: Dependency constraints over entities and their links.

$$\text{PMI}(L_i, L_j) = \log \frac{p(L_i, L_j)}{p(L_i)p(L_j)} \quad (4.1)$$

where $p(L_i)$ and $p(L_j)$ are the frequency of L_i and L_j respectively, and $p(L_i, L_j)$ is the co-occurrence frequency of $L_i(A, B)$ and $L_j(A, B)$ for any two entities A and B . Similarly, we apply a multivariate generalization form of PMI [107] to measure the triangle dependency:

$$\text{PMI}(L_i, L_j, L_k) = \log \frac{p(L_i, L_j, L_k)}{p(L_i)p(L_j)p(L_k)} \quad (4.2)$$

For location and GPE entities, we normalize fine-grained locations to country/region level. For instance, “Baghdad” and “Fallujah” are two cities in “Iraq”, therefore they are considered the same place when we calculate PMI. To this end, we use Freebase [108] to search for the country of locations such as cities and states. For ambiguous locations, we choose the most salient countries with largest populations. Finally if the PMI value is lower than a certain threshold (in our experiment we used -2.0 for pairwise and -3.0 for triangle), the links are considered as *incompatible* and used as a constraint for global inference.

In total we learned 34 pairwise constraints and 16 triangle constraints. Some examples are listed in Table 4.1. For instance, “Ariel Sharon” and “Mahmoud Abbas” are frequently involved in Contact.Meet events, so they are unlikely to be members of the same organization according to the pairwise constraint. If “Osama bin Laden” and “George W. Bush” are involved in a Conflict event with high confidence, then they are unlikely to be the members of the same organization according

Table 4.1: Examples of incompatible constraints.

Pairwise	L_i	L_j	
	Person <i>A</i> founded Organization <i>B</i>	Organization <i>B</i> hired Person <i>A</i>	
	Person <i>A</i> has a Business relation with Person <i>B</i>	Person <i>A</i> has a Person-Social relation (e.g. family) with Person <i>B</i>	
Triangle-Entity	L_i	L_j	
	Organization <i>A</i> is involved in a Justice/Conflict/Transaction event with Organization <i>B</i>	Person <i>C</i> is affiliated with or member of both Organization <i>A</i> and Organization <i>B</i>	
Triangle-Link	L_i	L_j	L_k
	Entity <i>A</i> is involved in a Transport event originated from Location <i>B</i>	Person <i>C</i> is affiliated with or member of Entity <i>A</i>	Person <i>C</i> is located in Location <i>B</i>

to the triangle-entity constraint. If “Washington” and “Iraq” are involved in a “Transport” event, then any member of “Washington” is unlikely to be located in “Iraq” according to the triangle-link constraint.

ILP Formulation

Motivated by the constrained conditional models [62, 63, 67, 68], we take the above constraints as hard constraints to perform the global inference. These constraints are designed to guarantee that the facts extracted from different documents are consistent with each other, hence weak predictions that violate the constraints can be filtered out.

Assuming we have a set of inter-dependent predicates $\mathcal{R} = \{r_i\}$ from a baseline extractor. Each unique predicate r_i is associated with a number of mentions $r_{i,j}$ with local confidence values $p_{i,j}$, where $p_{i,j} \in (0, 1]$. From cross-document point of view, a reliable output should have high local confidence value as well as high global frequency. By contrast, an invalid predicate often has low frequency, simply because some entities accidentally co-occur in mis-leading contexts. Based on this assumption, we introduce the following objective function to incorporate those two properties (we use $r_{i,j}$ to denote the j -th occurrence of a relation instance r_i , and $p_{i,j}$ to denote its local confidence value):

$$\text{maximize } \sum_{i=0}^N (x_i \cdot \sum_{j=0}^M p_{i,j}^\theta) \quad (4.3)$$

where x_i is a binary value :

$$x_i = \begin{cases} 1 & \text{if } r_i \text{ is selected in final output} \\ 0 & \text{if } r_i \text{ is removed in final output} \end{cases}$$

To guarantee x_i to be a binary value, the following constraint on x_i should be satisfied:

$$x_i \in \{0, 1\} \quad \forall x_i \quad (4.4)$$

where θ determines to which extent we penalize low confidence values. If θ equals 0 then any confidence value should be considered equally as 1. As θ grows, it gives more penalty to lower confidence values. When $\theta = 1$, p^θ equals p itself. We formulate the constraints described in section 4.1 as follows:

1. For constraints that involve three predicates, if x_a , x_b , and x_c violate one of the them, they must satisfy :

$$x_a + x_b + x_c \leq 2 \quad (4.5)$$

2. Similarly, for constraints that involve two predicates, if x_a and x_b violate one of them, they must satisfy:

$$x_a + x_b \leq 1 \quad (4.6)$$

Equation 4.3-4.6 all together constitute a Binary Linear Programming problem

(BLP, a special case of ILP in which all variables must be binary values):

$$\begin{aligned}
 & \text{maximize} && \sum_{i=0}^N (x_i \cdot \sum_{j=0}^M p_{i,j}^\theta) \\
 & \text{subject to} && x_i \in \{0, 1\} \quad \forall x_i \\
 & && \forall (x_a, x_b, x_c) \text{ violate one of constraints :} && (4.7) \\
 & && \quad \quad \quad x_a + x_b + x_c \leq 2 \\
 & && \forall (x_a \text{ and } x_b) \text{ violate one of constraints} \\
 & && \quad \quad \quad x_a + x_b \leq 1
 \end{aligned}$$

To solve this problem, we use a public available package `lp_solve`⁷, which implements the branch-and-bound algorithm.

4.2 Experiments

In this section we present the results of applying this joint inference method to improve cross-document information extraction. We use the data set from the DARPA GALE distillation task for our experiment, which contains 381,588 newswire documents. The baseline IE system extracted 18,386 person entities, 21,621 geopolitical entities and 18,792 organization entities. Table 4.2 shows the total number of extracted relations and events in different types. We asked two human annotators to evaluate the quality of *Family* and *Member_of* relations.

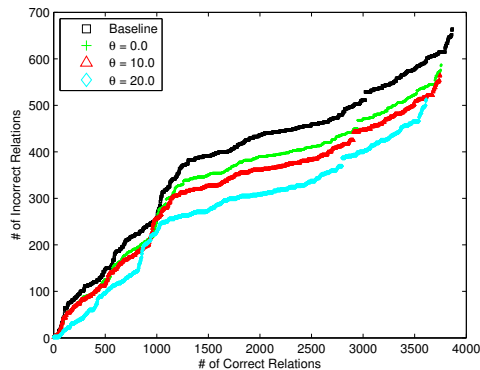
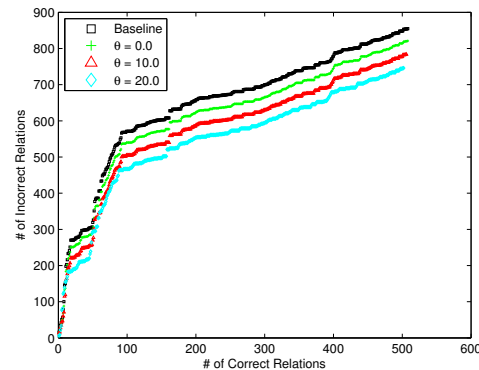
4.2.1 Overall Performance

Figure 4.2a and 4.2b demonstrate the browsing costs. Figure 4.3a and 4.3b depict the correctness of the removal operations (i.e., how many unique facts are removed correctly vs. incorrectly), varying the parameter θ of the objective function. We can see that compared to the baseline, our approach resulted in a 7.83%-29% user browsing effort reduction for *Family* relations and a 0.7%-32.3% user browsing effort reduction for *Member_of* relations. Although our method mistakenly removed a few correct facts, it successfully removed many more incorrect instances using any

⁷http://groups.yahoo.com/group/lp_solve (Date Last Accessed, March, 10, 2015)

Table 4.2: Numbers of unique relation and event predicates.

Relation/Event	ACE Definition	Number
Member Of	ORG-Aff.Employment, ORG-Aff.Membership	2,854
Family	Personal-Social.Family	1,128
Business	Personal-Social.Business	326
Entity_Located	Gen-Aff.Org-Location-Origin	7,504
Person_Located	Physical.Located	4,788
Residence	Gen-Aff.CRRE	2,560
Contact	Contact.Meet, Contact.Phone-Write	445
Transaction	Transaction.Transfer-Ownership, Transaction.Transfer-Money	345
Conflicts or Justice	Conflict, Justice, Life.Injure	976
Transport	Movement.Transport	338

(a) *Member_of* relation.(b) *Family* relation.**Figure 4.2: Browsing cost comparison of Member_of and Family relations.**

parameter. The overall rewards significantly outweigh the risks.

4.2.2 Impact of Different Types of Constraints

In order to evaluate the impact of each constraint, we also conducted experiments using each constraint independently in the ILP model with $\theta = 10$. The results are presented in Table 4.3.

The Triangle-Entity Compatible constraint aggressively removed many correct instances but also some incorrect ones. For example, the baseline IE system mistakenly predicted that “*Saddam Hussein*” as a member of “*Hezbollah Party*” and “*Amnesty International*”. This error can be removed based on the following

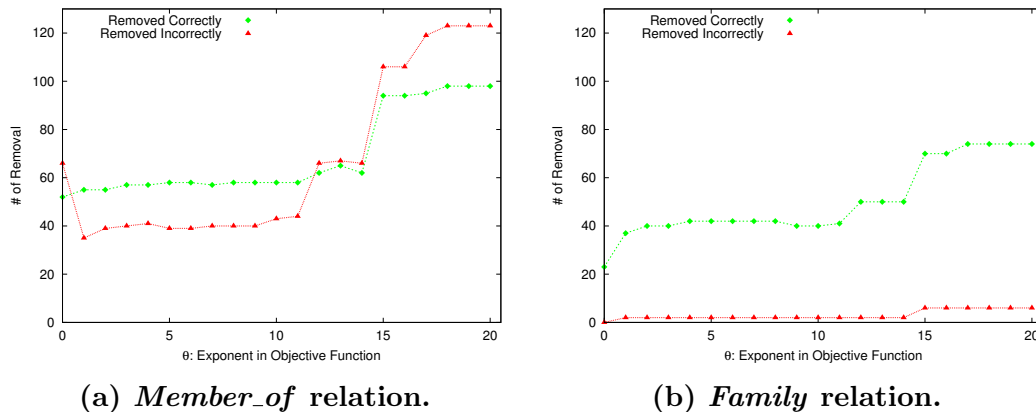


Figure 4.3: Removal curves w.r.t parameter θ . The x -axis is parameter θ 's value, y -axis is the number of relation instances removed.

Table 4.3: Impact of different constraints on Member_of and Family.

(a) *Member_of* Relation.

Constraint type	# removed correctly	# removed incorrectly
Pairwise	13	5
Triangle-Entity Compatible	83	56
Triangle-Entity Incompatible	22	11
Triangle-Link Incompatible	8	4

(b) *Family* Relation.

Constraint type	# removed correctly	# removed incorrectly
Pairwise	26	2
Triangle-Entity Compatible	46	1

high-confidence facts: “*Saddam Hussein*” lived in “*Tikrit*” of “*Iraq*”, but “*Hezbollah Party/Amnesty International*” were located in different regions “*Lebanon/UK*”. We can see that the pairwise constraint is very powerful, especially for the *Family* relation. For example, it removed the *Family* relation between “*Jack Straw*” and “*Tony Blair*” because they were involved in the *Family* relation (“*Jack Straw*” was in “*Tony Blair*”’s Cabinet). It occasionally removed a few correct relations involving two person entities with multiple types of relations. For example, “*Mohammed Bakir Al-hakim*” and “*Abdul Aziz Al-hakim*” are family members as well as colleagues in the Iraqi government.

Analysis

The underlying sentence-level IE system is far from perfect, and many spurious outputs maybe produced from various contexts. For instance:

Doc 1 *The list included Sheik Ahmed Yassin, Hamas' founder and spiritual leader, senior Hamas official Abdel Aziz Rantisi*

Doc 2 *Since the June 4 summit in Jordan between Abbas, Sharon and George W. Bush, Hamas has been a thorn in the side of Abbas ...*

In the first document, the underlying system successfully extracted the `Member_of` relation between “*Ahmed Yassin*” and “*Hamas*”, but in the second document, “*George W. Bush*” is mistakenly labeled as member of “*Hamas*” with fairly high confidence, because of a pattern for `Employment` relation: “`PER, ORG`”. with the help of relational dependencies among relations and events in cross-document level, such mistakes can be effectively recovered.

Due to the errors in the local name tagger and relation extractor, there may exist relation extraction output with incorrect argument types. For example, a baseline system may consider “*Lebanon*” as a person entity, thus incorrectly predict “*Lebanon*” as member of “*Hezbollah*” in a certain context. This kind of errors can be filtered out using this constraint, since the relation between “*Hezbollah*” and “*Lebanon*” should be detected correctly in most cases. In additional to the experimental results in the previous section, we are also curious about whether the proposed inference method can improve the results of the joint extraction framework in Chapter 3. We found that by using the constraints that we discovered in this method we can only remove very few false positive predictions from the joint model. For example, the model mistakenly considered that “*Israel*” is located in “*Gaza*”. Given that those two entities are frequently involved in transport events, the inference method can remove this predicate. The impact of the inference method on this result is insignificant since (i) our joint model does not produce any entity co-reference information, as a result, many predicates involve only nominal mentions and pronouns; and (ii) the test data is small, therefore there is not enough redundancy information.

4.3 Discussion

In this chapter we used an ILP-based inference framework to exploit cross-document dependencies over multiple facts extracted from a large set of documents. In practical application of information extraction, the documents of interests can be from different news agencies, different time frames and different genres. The knowledge that conveyed by those documents are redundant. Because of the difference among the contextual information, the extraction results from different places are often inconsistent. To improve the quality of extraction, we mined a set of pairwise and triangle constraints, and devised an ILP formulation to optimize the overall confidence subject to those constraints. Such joint inference analysis allowed us to significantly enhance the extraction performance.

CHAPTER 5

Joint Bilingual Name Tagging

In this chapter, we present the idea of utilizing cross-lingual dependencies to improve IE for parallel corpora. We propose to jointly and consistently extract names from parallel corpora by allowing interactions between the bilingual sentence pairs. Experiments on Chinese-English parallel corpora demonstrated that the proposed methods significantly outperformed monolingual baselines, and were robust to automatic word alignment. External evaluation on name-aware machine translation showed that the proposed name tagger can be applied to improve word alignment and name-aware machine translation.

5.1 Baseline Approach

Traditionally, name tagging is modeled as a sequential labeling problem, which takes input as a sequence of words/tokens, and predicts a chain of corresponding labels. In this study, we aim to address the problem of joint bilingual name tagging to extract names in parallel corpus coherently and accurately. In our case study, the input of a bilingual name tagger is aligned (manually or automatically) parallel sentence pairs in Chinese and English. We first apply the Stanford word segmenter [109] with Peking University standard to segment Chinese sentences. For example, for the parallel sentence pair demonstrated in Figure 5.1, our goal here is to extract name pairs that appear in the bilingual sentence pair, such as the organization name pair of (亞行, Asian Development Bank).

A natural and straightforward approach is to consider each side of a sentence pair in isolation, and solve the sequence labeling problem on each side. In post-processing, we can remove all of those name pairs that are mis-aligned in boundaries

Portions of this chapter previously appeared as: Q. Li, H. Li, H. Ji, W. Wang, J. Zheng, and F. Huang, “Joint Bilingual Name Tagging for Parallel Corpora,” in *Proc. Int. Conf. on Inform. and Knowledge Manage.*, Maui, HI, 2012, pp. 1727–1731.

Portions of this chapter previously appeared as: H. Li, J. Zheng, H. Ji, Q. Li, and W. Wang, “Name-aware machine translation,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Sofia, Bulgaria, 2013, pp. 604–614.

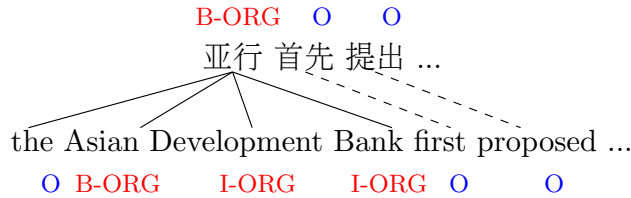


Figure 5.1: Example of parallel sentence pair. Solid lines represent alignments between names, while dashed lines denote other alignments

or labeled with different types. We adopt the linear-chain CRF [94] as our learning method. Given an input sequence \mathbf{x} , the conditional distribution of the output label sequence \mathbf{y} is defined as:

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \cdot \exp \sum_{j=1}^L \sum_{k=1}^K \theta_k \cdot f_k(y_j, y_{j-1}, \mathbf{x}, j) \quad (5.1)$$

where f_k is a feature function, θ_k is its weight, and $Z(\mathbf{x})$ is a normalization factor.

To cast name tagging as a sequential labeling problem, the BIO tagging scheme [31] is applied as our label alphabet. Each token in the input sequence is labeled with one of BIO tags in conjunction with its entity type. BIO means a token is Beginning of, Inside, and Out of a named entity, respectively. There are several out-of-shelf toolkits for linear-chain CRF, and we use Mallet [95] in our experiments. Table 5.1 summarizes the features for the baseline, where we assume the i -th token is the token in the current step.

5.2 Joint Bilingual Name Tagger

5.2.1 Linear-chain CRF with Cross-lingual Features

The baseline approach that we described above neglects the dependencies between sentence pairs. Following the intuition that the contexts of sentences pairs can help disambiguate and reduce errors mutually, we present a new approach that still takes linear-chain CRF as the learning framework, but exploits cross-lingual contexts based on alignments.

Let $\mathbf{x}_c = (x_{c,1} \dots x_{c,L})$ and $\mathbf{x}_e = (x_{e,1} \dots x_{e,M})$ be the input Chinese-English sentence pair; $\mathbf{y}_c = (y_{c,1} \dots y_{c,L})$ and $\mathbf{y}_e = (y_{e,1} \dots y_{e,M})$ be the corresponding output label sequences. The subscripts c and e denote Chinese and English respectively. In

Table 5.1: Monolingual features in baseline systems. The examples correspond to Figure 5.1. The token “Asian” of the English sentence is used as current token. The subscripts represent the offsets from the current token.

Language	Feature	Description
Common Language-independent	n-gram	Unigram, bigram and trigram token sequences in the context window of the current token. For example, $w_{-2}w_{-1}$ = “, the”; w_3 = “first”.
	Part-of-Speech	Part-of-Speech tags in the contexts are used. For example, $POS_1=N$.
	Dictionary	Various types of gazetteers, such as person names, organizations, countries and cities, titles and idioms are used. For example, a feature “B-Country” means the current token is the first token of an entry of our country name list.
	Conjunction	Conjunctions of various features. For example, $POS_1POS_2=N\&N$
English-specific	Brown Word Cluster	To reduce sparsity, we use the Brown clusters learned from ACE English corpus as features [99]. We use the clusters with prefixes of length 4, 6, 10 and 20.
	Case and Shape	English capitalization and morphology analysis based features. For example, “ <i>InitCap</i> ” indicates whether the token’s first character is capitalized.
	Chunking	Chunking tags are used as features. For example, $Chunk_1 = I.NP$.
	Other	Sentence level and document level features. For example, T_{FIRST} means the token is in the first sentence of a document.
Chinese-specific	Rule-based feature	Some heuristic rules are designed to detect person names using first name and last name character lists. For example, for a sequence of words, if all characters appear in the first name character list, and the length of each word is less than 2, then the sequence is likely to be a person’s first name.

Chinese each $x_{c,i}$ is a word, while in English each $x_{e,j}$ represents a token. We use $\mathcal{A} = \{(i, j)\}$ to denote the set of Chinese-English alignments, an alignment (i, j) indicates a Chinese word $x_{c,i}$ is aligned to an English token $x_{e,j}$. For simplicity we take Chinese side as an example, the hidden variables \mathbf{y}_c is not only conditioned on \mathbf{x}_c but also conditioned on \mathbf{x}_e and its alignment \mathcal{A} . The conditional probability of

\mathbf{y}_c can be extended as:

$$P(\mathbf{y}_c | \mathbf{x}_c, \mathbf{x}_e, \mathcal{A}) = \frac{1}{Z} \cdot \exp \sum_{i=1}^T \sum_{k=1}^K (\theta_k \cdot f_k(y_{c,i}, y_{c,i-1}, \mathbf{x}_c, \mathbf{x}_e, i, \mathcal{A}[i])) \quad (5.2)$$

where $\mathcal{A}[i]$ represents the indices of English tokens which are aligned to the i -th Chinese word. This still follows the linear-chain structure in which we need to build one model for each language. The distinction from the baseline approach is that, with an English sequence \mathbf{x}_e and its alignment \mathcal{A} , we can propagate the context from English to Chinese according to its alignment, and vice versa. Therefore not only is the output from two languages more accurate, but also the entity pair detection performance is improved consequently. Ideally, we can generate arbitrary variants from the feature function $f_k(y_{c,i}, y_{c,i-1}, \mathbf{x}_c, \mathbf{x}_e, i, \mathcal{A}[i])$. In practice, we use the same feature set as in Section 5.1, but aggregate features of $x_{e,j}$ and its corresponding English tokens as observed features.

5.2.2 Bilingual Conditional Random Fields

Although the approach in section 5.2.1 already takes into account the dependencies between sentence pair, it requires separate models for two languages, and the prediction from one side cannot directly influence the assignment of the other. In this section, we devise a bilingual CRF framework to jointly model the bilingual sentence pair based on their alignments. We define the conditional probability of output \mathbf{y}_c and \mathbf{y}_e jointly as:

$$p(\mathbf{y} | \mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{i=1}^L \Psi_c(y_{c,i}, y_{c,i-1}, \mathbf{x}_c, i) \prod_{j=1}^M \Psi_e(y_{e,j}, y_{e,j-1}, \mathbf{x}_e, j) \prod_{(i,j) \in \mathcal{A}} \Psi_a(y_{c,i}, y_{e,j}, \mathbf{x}_c, \mathbf{x}_e, i, j) \quad (5.3)$$

This distribution is factorized by three cliques of factors: $\{\Psi_c\}$ are potentials of Chinese linear-chain factors, $\{\Psi_e\}$ are potentials of English linear-chain factors, and $\{\Psi_a\}$ are potentials of bilingual factors. The factors in each clique share the same feature set and weights. $Z(\mathbf{x})$ is the normalization factor which sums over potentials of all possible assignments of \mathbf{y}_c and \mathbf{y}_e .

Monolingual Linear-chain Factors

Similar to monolingual name tagging, for any sentence in each language we define factors over all pairs of consecutive variables (y_{t-1}, y_t) , which enable the model to capture the dependency between consecutive variables. The potential function of monolingual factors Ψ_c and Ψ_e is defined as

$$\Psi(y_t, y_{t-1}, \mathbf{x}, t) = \exp\left(\sum_k \lambda_k f_k(y_t, y_{t-1}, \mathbf{x}, t)\right) \quad (5.4)$$

where f_k is a binary feature function and λ_k is the corresponding real-valued weight.

Bilingual Alignment Factors

The label of a Chinese word is often highly correlated with its aligned English token, and vice versa. For instance, in the example in Figure 5.1, the Chinese word “亚行” and its English counterpart “*Asian Development Bank*” should be both labeled as organizations. In order to model the correlation between the labels of aligned word-tokens, we introduce factors that link output variables in two languages based on alignments. For alignment (i, j) in which Chinese word $x_{c,i}$ is aligned to English token $x_{e,j}$, we define a bilingual factor over $y_{c,i}$ and $y_{e,j}$. This factor template bridges two monolingual linear chains, and makes it possible to propagate information across the sentences pairs. The potential function of bilingual factors Ψ_a is defined as:

$$\Psi_a(y_{c,i}, y_{e,j}, \mathbf{x}_c, \mathbf{x}_e, i, j) = \exp\left(\sum_k \lambda_k f_k(y_{c,i}, y_{e,j}, \mathbf{x}_c, \mathbf{x}_e, i, j)\right) \quad (5.5)$$

This allows us to design arbitrary binary features based on both \mathbf{x}_c and \mathbf{x}_e . A simple feature function for the above example is:

$$f_{101}(y_{c,i}, y_{e,j}, \mathbf{x}_c, \mathbf{x}_e, i, j) = \begin{cases} 1 & \text{if } y_{c,i} = \text{B-ORG}, y_{e,j} = \text{I-ORG} \text{ and } x_{e,j} = \text{Bank} \\ 0 & \text{otherwise} \end{cases}$$

This feature is true when the English token “Bank” is tagged as I-ORG, and its aligned Chinese word is tagged as B-ORG. If this feature attains high weight, the aligned word-token pair is likely to represent an organization entity given that the English token is “*Bank*”.

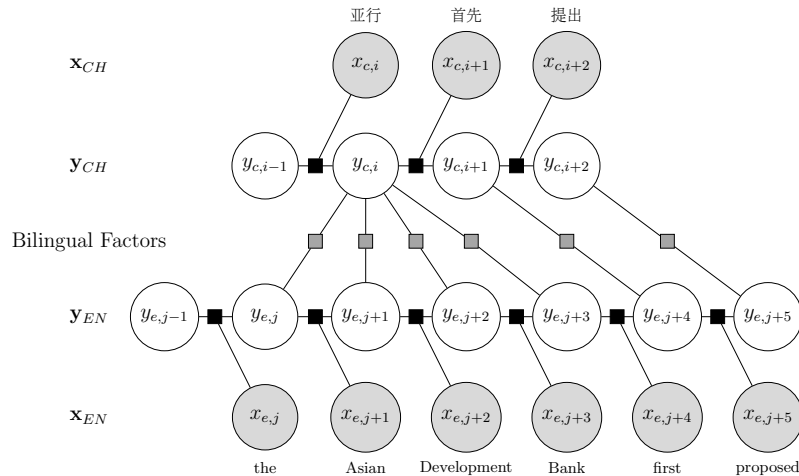


Figure 5.2: Graphical representation of bilingual CRF model. Squares represent factors over input and output variables, for simplicity, the links between bilingual factors and input variables are not shown.

Figure 5.2 illustrates the factor graph representation of the model for the example sentence pair in Figure 5.1. In this figure, white circles represent hidden variables \mathbf{y}_c and \mathbf{y}_e , gray circles represent observed sentence pair. Theoretically the factors can be linked to the whole observed sequences, for simplicity we only show the link to those at the same step.

Inference and Training

Since cycles are introduced by bilingual factors, typical inference algorithms for marginal probability and MAP such as Forward-backward and Viterbi algorithms cannot be exploited, and the exact inference is intractable in general. In this study we employ named Tree-Based Reparameterization (TRP) [110, 111], an efficient loopy belief propagation method, to perform approximate inference on the loopy graph. Given a set of training data $\{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^N$, the feature weights $\Lambda = \{\lambda_k\}$ are estimated using maximum likelihood estimation (MLE). The log-likelihood of the training set is calculated as:

$$\mathcal{L}_\Lambda = \sum_{i=1}^N \log p(\mathbf{y}^{(i)} | \mathbf{x}^{(i)}; \Lambda) \quad (5.6)$$

To avoid over-fitting we introduce Gaussian prior $\frac{|\Lambda|^2}{2\sigma^2}$ as regularization term to \mathcal{L}_Λ . Then the partial derivative of the log-likelihood with respect to parameter λ_k is:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \lambda_k} &= \sum_{i=1}^N F_k(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) \\ &\quad - \sum_{i=1}^N \sum_{\mathbf{y}'} p(\mathbf{y}' | \mathbf{x}^{(i)}; \Lambda) F_k(\mathbf{x}^{(i)}, \mathbf{y}') - \frac{\lambda_k}{\sigma^2} \end{aligned} \quad (5.7)$$

where $F_k(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$ denotes the count of feature f_k over the i -th instance. The first term is the empirical count of λ_k , and the second term is the expected count of λ_k under the model distribution. Given the gradient, optimization algorithms such as L-BFGS can be applied to maximize the log-likelihood. In this study we use Mallet [95] toolkit to implement the inference and learning process.

Features

Given such a framework, the remaining challenge is to design features for both monolingual and bilingual factors. There are many possible ways to define cross-lingual features in this joint model. For instance, one possibility is to define them based on some conjunctions of the observed values from two languages, but such features require very large of training data and thus suffer from data sparsity. In our framework, each feature is defined as a conjunction of assignment and features from the input sequence; therefore we only need to design features of the input sequence. We use the features presented in Section 5.1 for monolingual factors. The features for the proposed bilingual factors are based on the combination of the monolingual features from the corresponding words/tokens. For instance, given a bilingual factor over $x_{c,i}$ and $x_{e,j}$ and alignment (i, j) , the sets of monolingual features from $x_{c,i}$ and $x_{e,j}$ are merged as features to form the factor. In this way, both monolingual features and cross-lingual transferred features are incorporated in a uniformed manner.

5.3 Experiments

5.3.1 Evaluation Setup

We asked four bilingual speakers to manually annotate the Parallel Treebank, which contains 288 Chinese-English parallel documents aligned at token level man-

Table 5.2: The number of names in the bilingual data set.

Type	English	Chinese	Bilingual Pairs
GPE	4,049	4,077	4,031
PER	1,053	1,048	1,044
ORG	1,547	1,549	1,541
All	6,649	6,674	6,616

ually. The manual annotations were reviewed and improved with several additional passes to form the final ground-truth. 230 documents are randomly selected for training, and the remaining 58 documents are used for blind test. Some statistics about this bilingual data set are given in Table 5.2. The last column (Bilingual Pairs) of the table shows the number of name pairs detected with manual alignment. Since the translation is not exactly literal, some names in one language may have no correspondences in the other. As a result, the number of name pairs may be slightly smaller than the number of names in each language.

A name pair in system output is considered correct if and only if both names in both languages are correct and have the same entity type. The scores are computed using bilingual sentence pairs and name pairs, which are detected according to token-based alignment.

5.3.2 Overall Performance

Table 5.3 shows the proposed approaches (with both manual alignment and automatic alignment [112]⁸) substantially outperformed the baseline on all name types, at 99.9% confidence level according to Wilcoxon Matched-Pairs Signed-Ranks Test. The joint model achieved the top F-score with automatic alignment for organization names. This result indicates that our joint methods are robust to alignment noise and thus they can be practically applied to bilingual parallel data with automatic alignment, and alleviate the necessity of costly manual alignment.

Table 5.3: Performance (%) on bilingual data set. The bold F-scores are significantly better than the baseline; while the scores marked with * are the best for each type.

Method	Type	Bilingual Name Pair Tagging			
		GPE	PER	ORG	ALL
Baseline	P	89.2	91.3	68.9	85.9
	R	86.7	81.8	51.1	78.2
	F	87.9	86.3	58.7	81.9
Linear-Chain CRF with Cross-lingual Features (Manual Alignment)	P	91.2	92.4	69.8	87.2
	R	91.4	89.1	61.5	84.5
	F	90.7	91.2	65.4	85.8
Joint CRF (Manual Alignment)	P	90.8	94.0	68.6	86.9
	R	92.8	90.1	61.5	85.6
	F	91.8*	92.0*	64.9	86.3*
Linear-Chain CRF with Cross-lingual Features (Automatic Alignment)	P	90.6	97.0	70.7	87.8
	R	88.8	83.9	57.6	81.3
	F	89.7	90.0	63.5	84.4
Joint CRF (Automatic Alignment)	P	89.9	92.6	71.2	86.6
	R	88.7	84.9	61.9	82.3
	F	89.3	88.6	66.2*	84.4
Human Annotator	F	95.5	89.9	93.8	94.1

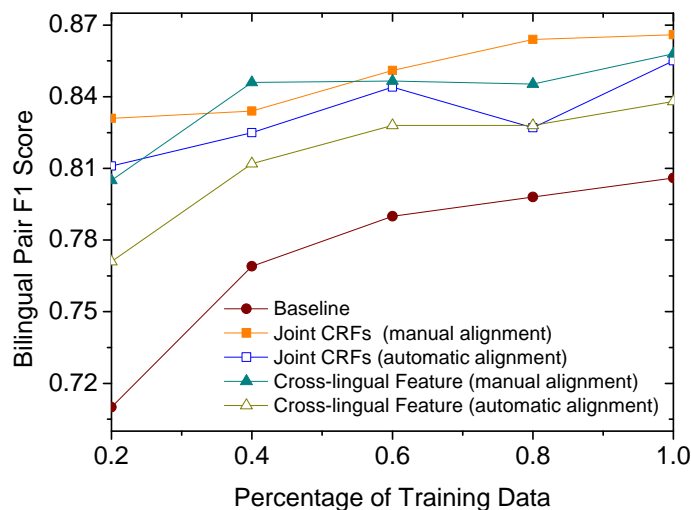


Figure 5.3: Performance on different sizes of training data.

5.3.3 Learning Curves

Figure 5.3 shows the overall performance of our models when they are learned from different size of training data. In order to balance the small size of training

⁸We applied GIZA++ 2.0 toolkit to produce automatic word alignment; default parameter setting for training, 5 iterations of IBM model 1, 3, 4 and HMM alignment model were performed respectively; the alignment f-measure was 56.7%

data and test data, we randomly selected half of the test set (29 documents) for test. We can see that with only 20% of the training data, each of the joint methods (with manual alignment or automatic alignment) can already achieve better performance compared to the baseline learned from 100% training data. In particular, when using 20% training data, the joint CRF model obtained 12.1% higher F-score with manual alignment and 10.1% higher F-score with automatic alignment over the baseline. As the training size increases, they consistently outperformed the baseline.

5.4 Extrinsic Evaluation on Name-aware Machine Translation

One ambitious goal of developing the bilingual name tagging system is to improve machine translation performance with respect to named entities. Traditional MT approaches focused on the fluency and accuracy of the overall translation but lack their ability to translate certain content words including critical information, especially names. A typical statistical MT system can only translate 60% person names correctly [113]. Incorrect segmentation and translation of names, which often carry central meanings of a sentence, can also yield incorrect translation of long contexts. We developed a novel Name-aware MT (NAMT) approach which can tightly integrate our bilingual name tagging into the training and decoding processes of an end-to-end MT pipeline.

As our baseline, we apply a high-performance Chinese-English MT system [114, 115] based on hierarchical phrase-based translation framework [116]. It is based on a weighted synchronous context-free grammar (SCFG). In our NAMT approach, we apply our bilingual name tagger to extract three types of names: `PER`, `ORG` and `GPE` from both the source side and the target side in a parallel corpus with word alignment from running GIZA++ [117]. We ignore two kinds of names: multi-word names with conflicting boundaries in two languages and names only identified in one side of a parallel sentence.

We built a NAMT system from such name-tagged parallel corpora. First, we replace name pairs with their entity types, and then use GIZA++ to re-generate word alignment. Since the name tags appear very frequently, the existence of such

tags yields improvement in word alignment quality. The re-aligned parallel corpora are used to train our NAMT system based on SCFG. We extract SCFG rules by treating the tagged names as non-terminals. However, the original parallel corpora contain many high-frequency names, which can already be handled well by the baseline MT. Replacing them with non-terminals can lead to information loss and weaken the translation model. To address this issue, we merged the name-replaced parallel data with the original parallel data and extract grammars from the combined corpus. For example, given the following sentence pair:

- 中国 反对 外来 势力 介入 安哥拉 冲突 .
- China appeals to world for non involvement in Angola conflict .

after name tagging it becomes

- GPE 反对 外来 势力 介入 GPE 冲突 .
- GPE appeals to world for non involvement in GPE conflict .

Both sentence pairs are kept in the combined data to build the translation model. During the decoding, we extract names in the source language with the baseline monolingual name tagger. Then we apply a state-of-the-art name translation system [113] to translate names into the target language. The non-terminals in SCFG rules are rewritten to the extracted names during decoding, therefore allow unseen names in the test data to be translated. Finally, our decoder exploits the dynamically created phrase table from name translation, competing with originally extracted rules, to find the best translation for the input sentence.

5.4.1 Evaluation Setup

We used a large Chinese-English MT training corpus from various sources and genres (including newswire, web text, broadcast news and broadcast conversations) for our experiments. We also used some translation lexicon data and Wikipedia translations. The training corpus includes 1,686,458 sentence pairs. The joint name tagger extracted 1,890,335 name pairs (295,087 PER, 1,269,056 GPE and 326,192 ORG).

5.4.2 Overall Performance

We evaluate the overall performance using the following three evaluation metrics (see [21] for details):

1. Name-aware BLEU metric: $\text{BLEU}_{\text{NA}} = \text{BP} \cdot \text{NP} \cdot \exp\left(\sum_{n=1}^N w_n \log wp_n\right)$. This metric is augmented from original BLEU metric [118]. Each word count in the brevity penalty BP is weighted based on whether it is contained in a name. NP is to penalize the output sentences that contain too many or too few names. wp_n is weighted precision.
2. Translation Edit Rate (TER) [119]. $\text{TER} = \frac{\# \text{ of edits}}{\text{average } \# \text{ of reference words}}$.
3. Named Entity Weak Accuracy (NEWA) [17] to evaluate the name translation performance. $\text{NEWA} = \frac{\text{Count } \# \text{ of correctly translated names}}{\text{Count } \# \text{ of names in references}}$.

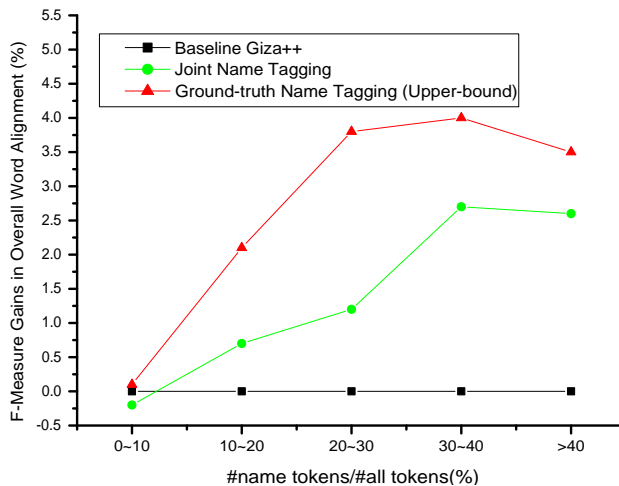
Besides the original baseline, we developed another baseline system by adding name translation table into the phrase table (NPhrase). Table 5.4 presents the performance of overall translation and name translation. We can see that except for the BOLT3 data set with BLEU metric, our NAMT approach consistently outperformed the baseline system for all data sets with all metrics, and provided up to 23.6% relative error reduction on name translation. According to Wilcoxon Matched-Pairs Signed-Ranks Test, the improvement is not significant with BLEU metric, but is significant at 98% confidence level with all of the other metrics. The gains are more significant for formal genres than informal genres mainly because most of the training data for name tagging and name translation were from newswire. Furthermore, using external name translation table only did not improve translation quality in most test sets except for BOLT2. Therefore, it is important to use name-replaced corpora for rule extraction to fully take advantage of improved word alignment.

5.4.3 Improving Word Alignment

It is also interesting to examine the advantage of our bilingual name tagging on improving word alignment. We conducted the experiment on the Chinese-English Parallel Treebank [120] with ground-truth word alignment. The detailed procedure is as follows:

Table 5.4: Translation performance (%).

Metric	System	BOLT 1	BOLT 2	BOLT 3	BOLT 4	BOLT 5	NIST2006	NIST2008	
BLEU	Baseline	14.2	14.0	17.3	15.6	15.3	35.5	29.3	
	NPhrase	14.1	14.4	17.1	15.4	15.3	35.4	29.3	
	NAMT	14.2	14.6	16.9	15.7	15.5	36.3	30.0	
Name-aware BLEU	Baseline	18.2	17.9	18.6	17.6	18.3	36.1	31.7	
	NPhrase	18.1	18.8	18.5	18.1	18.0	35.8	31.8	
	NAMT	18.4	19.5	19.7	18.2	18.9	39.4	33.1	
TER	Baseline	70.6	71.0	69.4	70.3	67.1	58.7	61.0	
	NPhrase	70.6	70.4	69.4	70.4	67.1	58.7	60.9	
	NAMT	70.3	70.2	69.2	70.1	66.6	57.7	60.5	
NEWA	All	Baseline	69.7	70.1	73.9	72.3	60.6	66.5	60.4
		NPhrase	69.8	71.1	73.8	72.5	60.6	68.3	61.9
		NAMT	71.4	72.0	77.7	75.1	62.7	72.9	63.2
	GPE	Baseline	72.8	78.4	80.0	78.7	81.3	79.2	76.0
		NPhrase	73.6	79.3	79.2	78.9	82.3	82.6	79.5
		NAMT	74.2	80.2	82.8	80.4	79.3	85.5	79.3
	PER	Baseline	53.3	44.7	45.1	49.4	48.9	54.2	51.2
		NPhrase	52.2	45.4	48.9	48.5	47.6	55.1	50.9
		NAMT	55.6	45.4	58.8	55.2	56.2	60.0	52.3
	ORG	Baseline	56.0	49.0	52.9	38.1	41.7	44.0	41.3
		NPhrase	50.5	50.3	54.4	40.7	41.3	42.2	40.7
		NAMT	60.4	52.3	55.4	41.6	45.0	51.0	44.8

**Figure 5.4: Word alignment gains according to the percentage of name words in each sentence.**

1. run the joint bilingual name tagger,
2. replace each name string with its name type (PER, ORG or GPE), and ran GIZA++ on the replaced sentences,
3. run GIZA++ on the words within each name pair,
4. and merge the results from (2) and (3) as the final word alignments.

Table 5.5: Impact of joint bilingual name tagging on word alignment.

Words	Method	P (%)	R (%)	F (%)
All Words	Baseline GIZA++	69.8	47.8	56.7
	Joint Name Tagging	70.4	48.1	57.1
	Ground-truth Name Tagging	71.3	48.9	58.0
Words Within Names	Baseline GIZA++	86.0	31.4	46.0
	Joint Name Tagging	77.6	37.2	50.3

We also measured the performance of applying ground-truth named entities as the upper-bound. The experiment results are shown in Table 5.5. For the words within names, our approach provided significant gains by enhancing F-measure from 46.0% to 50.3%. Only 10.6% words are within names, therefore the upper-bound gains on overall word alignment is only 1.3%. Our joint name tagging approach achieved 0.4% (statistically significant) improvement over the baseline. In Figure 5.4 we categorized the sentences according to the percentage of name words in each sentence and measured the improvement for each category. We can clearly see that as the sentences include more names, the gains achieved by our approach tend to be greater.

5.4.4 Analysis

Although the proposed model has significantly enhanced translation quality, some challenges remain. Here we highlights some major problems.

Name Structure Parsing We found that the gains of our NAMT approach were mainly achieved for names with one or two components. When the name structure becomes too complicated to parse, name tagging and name translation are likely to produce errors, especially for long nested organizations. For example, “古田县 检察院 反渎局” (Anti-malfeasance Bureau of Gutian County Procuratorate) consists of a nested organization name with a GPE as modifier: “古田县 检察院” (Gutian County Procuratorate) and an ORG name: “反渎局” (Anti-malfeasance Bureau).

Name Abbreviation Tagging and Translation Some organization abbreviations are also difficult to extract because our name taggers have not incorporated any coreference resolution techniques. For example, without knowing that “FAW” refers to “*First Automotive Works*” in “*FAW has also utilized the capital market to directly finance, and now owns three domestic listed companies*”, our system mistakenly

labeled it as a GPE. The same challenge exists in name alignment and translation (for example, “民革 (Min Ge)” refers to “中国国民党革命委员会” (Revolutionary Committee of the Chinese Kuomintang)).

English Organization Tagging Sometimes the joint model cannot improve English Organization extraction. Some government organizations were only partially translated. For example, “柬埔寨王国政府 (*the Kingdom of Cambodia government*)” was translated and aligned to “*the Kingdom of Cambodia*”, where “*government*” was missed in English. In order to produce consistent name boundaries, the joint model mistakenly labeled “*the Kingdom of Cambodia*” as an English organization name. English monolingual features normally generate higher confidence than Chinese features for ORG names. On the other hand, some good propagated Chinese features were not able to correct English results. For example, in the following sentence pair: “根据中国，老挝和联合国难民署三方达成的... (*in accordance with the tripartite agreement reached by China, Laos and the UNHCR on*)...”, even though the tagger can successfully label “联合国难民署/*UNHCR*” as an organization because it is a common Chinese name, English features based on previous GPE contexts still incorrectly predicted “*UNHCR*” as a GPE name.

5.5 Discussion

In this chapter, we studied the cross-lingual dependencies in parallel corpora. We proposed a bilingual joint graphical model to improve name tagging performance on parallel corpora. Taking an English-Chinese parallel corpus as a case study, we demonstrated that this method significantly improves the traditional monolingual baseline. And the bilingual factors based on word alignments can encourage consistency between the sentence pairs. In addition, our external evaluation showed that the new method can be applied to improve name-aware machine translation and statistical word alignment.

CHAPTER 6

Conclusions and Future Directions

In this thesis we made use of *cross-component*, *cross-document*, and *cross-lingual* dependencies to improve IE performance. As the main part of this thesis, we presented a novel joint framework based on structured prediction and inexact search to extract entity mentions, relations and events from each sentence. Our main argument is that these fundamental IE subtasks should not be simply modeled by a pipeline of local classifiers as in most previous approaches. Instead, it is important to make use of interactions among different subtasks, and take into account non-local features that can capture various long-distance dependencies. Different from traditional pipelined approaches, for the first time, we formulated the task of IE as constructing information networks from input sentences, where our goal is to search for the best information network from each input sentence. Based on this powerful framework, we can extract multiple IE components such as entity mentions, relations and events all together in a single model. In addition, we have explored a number of useful global features to encourage global coherency. Our final system can replace a pipeline of multiple classifiers by a single and unified model to extract the three types of component simultaneously while achieving state-of-the-art performance in each subtask. Beyond the sentence-level extraction, we further presented a cross-document inference method based on ILP formulation, where the inter-dependencies among facts that extracted from different places are leveraged as hard constraints. This method allows us to incorporate information from a much wider context to further improve the extraction quality from a sentence-level extractor. Finally, we presented a bilingual name tagging framework to make use of bilingual dependencies in parallel corpora. Experiments on English-Chinese parallel corpus demonstrated that the joint bilingual model can produce more accurate and coherent extraction results comparing with the isolated monolingual baselines.

There are two main directions for future work:

A. Expanding Information Types We are interested in expanding the scope of extraction to explore more information types. So far, the sub-tasks that we have discussed in this thesis only cover a limited number of types. For example, the entity mention extraction only uses 7 main entity types in the ACE definition such as **Person**, **Organization**, **Geo-political Entity** and so on. The event extraction uses 33 event subtypes. Although the total number is much more than the entity types, it neglects a lot of useful and common event types in daily life. For example, consider the following news title about Ebola in New York:

“A Ebola patient in New York is cured and released from hospital.”

If we only use the types that were studied in this thesis, we can only find minimal information about the news conveyed by this title. To be specific, we cannot recognize “*Ebola*” as a disease, “*patient*” as a person entity who is under treatment for a medical problem. Moreover, the main events “*Cure* (cured)” and “*Discharge* (released)” are out of our domain of event mentions, but they play a critical role in this sentence. In another example:

“the Seattle Seahawks won the Super Bowl in a nearly flawless performance.”

only recognizing “Seattle Seahawks” as an Organization mention is not helpful to understand this sentence. It is important to extract the main event “*Win* (won)” and its arguments “*Sports Game* (Super Bowl)”, and “*Sports Team* (Seattle Seahawks)”. In addition, the fine-grained types can be utilized to further constrain the search space of extraction. For instance, we can say that sports team is the only type of organization that can participate in “*Win*” events. This future work requires an extensive expansion of task definition with linguistic and practical motivations, as well as a large corpus with desired annotations. Techniques such as distant supervision [121], automatic taxonomy induction [122] and semi-supervised learning [123] may be applied to help reduce the cost.

B. Knowledge Acquisition for Information Extraction Our error analysis on the experiment results of the joint models calls for feeding common knowledge to the course of extraction. Most features that we developed in this thesis are based on lexical, syntactic, and shallow semantic resources. Various types of world knowl-

edge have been shown useful to IE tasks [9,124,125]. Most prior work, however, only discovered some narrow aspects of world knowledge to some specific tasks. We believe that systematic and comprehensive acquisition of world knowledge is extremely difficult but essential to improve the performance of IE to a completely new level. Here we describe a couple of examples based on our analysis.

- In the real world, each type of entity has a finite set of common attributes. Knowing this information, we can constrain our extraction model to a certain sub search space. For instance, in the sentence “*any use of chemical weapons would be counterproductive to Saddam*”, the word “use” should be identified as a trigger for **Attack** event because “chemical weapons” is to used for attacking someone. On the contrary, in the sentence “the hardware is used by the industry today.”, “use” is not an **Attack** trigger.
- Usually there are many different ways to express the same event. On the other hand, many quite different events can be classified to the same type. How to differentiate the subtle difference of the same word in different scenes is a challenging. Consider the following sentences:

- “*an al-Qaeda member was captured.*”
- “*the Italian ship was captured by Palestinian terrorists.* ”

While the most frequent **Arrest-Jail** event trigger is “arrest”, the word “capture” in the first sentence has very similar meaning since it has a sense of “to take control of or seize by force”. In the second sentence, it should be a **Transfer-Ownership** trigger, since its object is an artifact rather than a person, and thus the sense of “capture” has been subtly changed.

As the final remark, although the task of IE has gained significant improvement over the last decade, the IE techniques are yet far from perfect. This thesis, by standing upon the shoulders of previous research such as constrained conditional models and structured prediction, studied the topic of joint information extraction for entity mentions extraction, relation extraction and event extraction, and provided a novel view for the whole task. We hope the results of this thesis can inspire further research in the field of IE and related areas.

REFERENCES

- [1] V. Ng, “Supervised noun phrase coreference research: The first fifteen years,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Uppsala, Sweden, 2010, pp. 1396–1411.
- [2] N. A. Chinchor, “Overview of muc-7/met-2,” in *Proc. Message Understanding Conf. MUC-7*, Fairfax, VA, 1999, pp. 1–4.
- [3] H. Ji and R. Grishman, “Knowledge base population: Successful approaches and challenges,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Portland, OR, 2011, pp. 1148–1158.
- [4] D. Nadeau and S. Sekine, “A survey of named entity recognition and classification,” *Named Entities: Recognition, Classification and Use. Special Issue of Lingvisticae Investigationes*, vol. 30, no. 1, pp. 3–26, Jan. 2007.
- [5] “Ace (automatic content extraction) english annotation guidelines for events,” Linguistic Data Consortium, Jan. 2005. [Online]. Available: <https://www ldc.upenn.edu/collaborations/past-projects/ace> (Date Last Accessed, March, 10, 2015)
- [6] S. Liao and R. Grishman, “Using document level cross-event inference to improve event extraction,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Uppsala, Sweden, 2010, pp. 789–797.
- [7] H. Ji and R. Grishman, “Refining event extraction through cross-document inference,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Columbus, OH, 2008, pp. 254–262.
- [8] Y. Hong, J. Zhang, B. Ma, J.-M. Yao, G. Zhou, and Q. Zhu, “Using cross-entity inference to improve event extraction,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Portland, OR, 2011, pp. 1127–1136.
- [9] Y. Chan and D. Roth, “Exploiting background knowledge for relation extraction,” in *Proc. Int. Conf. on Computational Linguistics*, Beijing, China, 2010, pp. 152–160.
- [10] Y. Chan and D. Roth, “Exploiting syntactico-semantic structures for relation extraction,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Portland, OR, 2011, pp. 551–560.

- [11] G. Zhou, J. Su, J. Zhang, and M. Zhang, “Exploring various knowledge in relation extraction,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Ann Arbor, MI, 2005, pp. 427–434.
- [12] R. Grishman, D. Westbrook, and A. Meyers, “Nyu english ace 2005 system description,” in *Proc. ACE 2005 Evaluation Workshop*, Gaithersburg, MD, 2005.
- [13] F. Smith, *Understanding Reading*. New York, NY: HRW, 1971.
- [14] T. Griffiths, N. Chater, C. Kemp, A. Perfors, and J. Tenenbaum, “Probabilistic models of cognition: Exploring representations and inductive biases,” *Trends in Cognitive Sciences*, vol. 14, no. 8, pp. 357–364, Aug. 2010.
- [15] D. Feng, Y. Lv, and M. Zhou, “A new approach for english-chinese named entity alignment,” in *Proc. Conf. on Empirical Methods in Natural Language Process.*, Barcelona, Spain, 2004, pp. 372–379.
- [16] Y. Deng and Y. Gao, “Guiding statistical word alignment models with prior knowledge,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Prague, Czech Republic, 2007, pp. 1–8.
- [17] U. Hermjakob, K. Knight, and H. D. III, “Name translation in statistical machine translation: Learn when to transliterate,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Columbus, OH, 2008, pp. 389–397.
- [18] M. Snover, X. Li, W.-P. Lin, Z. Chen, S. Tamang, M. Ge, A. Lee, Q. Li, H. Li, S. Anzaroot, and H. Ji, “Cross-lingual slot filling from comparable corpora,” in *Proc. 4th Workshop on Building and Using Comparable Corpora*, Portland, OR, 2011, pp. 110–119.
- [19] H. Ji and R. Grishman, “Analysis and repair of name tagger errors,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Sydney, Australia, 2006, pp. 420–427.
- [20] Q. Li, H. Li, H. Ji, W. Wang, J. Zheng, and F. Huang, “Joint Bilingual Name Tagging for Parallel Corpora,” in *Proc. Int. Conf. on Inform. and Knowledge Manage.*, Maui, HI, 2012, pp. 1727–1731.
- [21] H. Li, J. Zheng, H. Ji, Q. Li, and W. Wang, “Name-aware machine translation,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Sofia, Bulgaria, 2013, pp. 604–614.
- [22] Q. Li, S. Anzaroot, W.-P. Lin, X. Li, and H. Ji, “Joint inference for cross-document information extraction,” in *Proc. Int. Conf. on Inform. and Knowledge Manage.*, Glasgow, UK, 2011, pp. 2225–2228.

- [23] Q. Li, H. Ji, and L. Huang, “Joint event extraction via structured prediction with global features,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Sofia, Bulgaria, 2013, pp. 73–82.
- [24] Q. Li and H. Ji, “Incremental joint extraction of entity mentions and relations,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Baltimore, MA, 2014, pp. 402–412.
- [25] Q. Li, H. Ji, Y. HONG, and S. Li, “Constructing information networks using one single model,” in *Proc. Conf. on Empirical Methods in Natural Language Process.*, Doha, Qatar, 2014, pp. 1846–1851.
- [26] L. Chiticariu, Y. Li, and F. R. Reiss, “Rule-based information extraction is dead! long live rule-based information extraction systems!” in *Proc. Conf. on Empirical Methods in Natural Language Process.*, Seattle, WA, 2013, pp. 827–832.
- [27] D. Bikel, S. Miller, R. Schwartz, and R. Weischedel, “Nymble: a high-performance learning name-finder,” in *Proc. Conf. on Applied Natural Language Process.*, DC, 1997, pp. 194–201.
- [28] A. Borthwick, J. Sterling, E. Agichtein, and R. Grishman, “Exploiting diverse knowledge sources via maximum entropy in named entity recognition,” in *Proc. 6th Workshop on Very Large Corpora*, Montreal, Canada, 1998.
- [29] R. Florian, H. Hassan, A. Ittycheriah, H. Jing, N. Kambhatla, X. Luo, N. Nicolov, and S. Roukos, “A statistical model for multilingual entity detection and tracking,” in *Proc. Conf. of the North Amer. Chapter of the Assoc. for Computational Linguistics*, Boston, MA, 2004, pp. 1–8.
- [30] A. McCallum and W. Li, “Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons,” in *Proc. Conf. on Natural Language Learning*, Edmonton, Canada, 2003, pp. 188–191.
- [31] L. Ratinov and D. Roth, “Design challenges and misconceptions in named entity recognition,” in *Proc. Conf. on Natural Language Learning*, Boulder, CO, 2009, pp. 147–155.
- [32] J. R. Finkel, T. Grenager, and C. Manning, “Incorporating non-local information into information extraction systems by gibbs sampling,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Ann Arbor, MI, 2005, pp. 363–370.
- [33] V. Krishnan and C. D. Manning, “An effective two-stage model for exploiting non-local dependencies in named entity recognition,” in *Proc.*

- Annu. Meeting of the Assoc. for Computational Linguistics*, Sydney, Australia, 2006, pp. 1121–1128.
- [34] N. Kambhatla, “Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Barcelona, Spain, 2004, pp. 178–181.
- [35] S. Zhao and R. Grishman, “Extracting relations with integrated information using kernel methods,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Ann Arbor, MI, 2005, pp. 419–426.
- [36] R. C. Bunescu and R. J. Mooney, “A shortest path dependency kernel for relation extraction,” in *Proc. Conf. on Empirical Methods in Natural Language Process.*, Vancouver, Canada, 2005, pp. 724–731.
- [37] G. Zhou, M. Zhang, D.-H. Ji, and Q. Zhu, “Tree kernel-based relation extraction with context-sensitive structured parse tree information,” in *Proc. Conf. on Empirical Methods in Natural Language Process.*, Prague, Czech Republic, 2007, pp. 728–736.
- [38] F. Reichartz, H. Korte, and G. Paass, “Composite kernels for relation extraction,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Suntec, Singapore, 2009, pp. 365–368.
- [39] J. Jiang and C. Zhai, “A systematic exploration of the feature space for relation extraction,” in *Proc. Conf. of the North Amer. Chapter of the Assoc. for Computational Linguistics*, Rochester, NY, 2007, pp. 113–120.
- [40] D. Ahn, “The stages of event extraction,” in *Proc. Workshop on Annotating and Reasoning about Time and Events*, Sydney, Australia, 2006, pp. 1–8.
- [41] Z. Chen and H. Ji, “Language specific issue and feature exploration in chinese event extraction,” in *Proc. Conf. of the North Amer. Chapter of the Assoc. for Computational Linguistics*, Boulder, CO, 2009, pp. 209–212.
- [42] C. Chen and V. NG, “Joint modeling for chinese event extraction with rich linguistic features,” in *Proc. Int. Conf. on Computational Linguistics*, Mumbai, India, 2012, pp. 529–544.
- [43] R. C. Moore, “Learning translations of named-entity phrases from parallel corpora,” in *Proc. Conf. of the European Chapter of the Assoc. for Computational Linguistics*, Budapest, Hungary, 2003, pp. 259–266.
- [44] F. Huang and S. Vogel, “Improved named entity translation and bilingual named entity extraction,” in *Proc. ACM Int. Conf. on Multimodal Interaction*, Pittsburgh, PA, 2002, pp. 253–258.

- [45] Y. Chen, C. Zong, and K.-Y. Su, “On jointly recognizing and aligning bilingual named entities,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Uppsala, Sweden, 2010, pp. 631–639.
- [46] H. Ji and R. Grishman, “Collaborative entity extraction and translation,” in *Proc. Recent Advances of Natural Language Process.*, Borovets, Bulgaria, 2007, pp. 73–84.
- [47] S. Patwardhan and E. Riloff, “A unified model of phrasal and sentential evidence for information extraction,” in *Proc. Conf. on Empirical Methods in Natural Language Process.*, Suntec, Singapore, 2009, pp. 151–160.
- [48] S. Riedel and A. McCallum, “Fast and robust joint models for biomedical event extraction,” in *Proc. Conf. on Empirical Methods in Natural Language Process.*, Edinburgh, UK, 2011, pp. 1–12.
- [49] S. Riedel, H.-W. Chun, T. Takagi, and J. Tsujii, “A markov logic approach to bio-molecular event extraction,” in *Proc. BioNLP 2009 Workshop Companion Volume for Shared Task*, Boulder, CO, 2009, pp. 41–49.
- [50] D. McClosky, M. Surdeanu, and C. D. Manning, “Event extraction as dependency parsing,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Portland, OR, 2011, pp. 1626–1635.
- [51] S. Riedel and A. McCallum, “Robust biomedical event extraction with dual decomposition and minimal domain adaptation,” in *Proc. BioNLP Shared Task 2011 Workshop*, Portland, OR, 2011, pp. 46–50.
- [52] R. Florian, H. Jing, N. Kambhatla, and I. Zitouni, “Factorizing complex models: A case study in mention detection,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Sydney, Australia, 2006, pp. 9–16.
- [53] R. Florian, J. F. Pitrelli, S. Roukos, and I. Zitouni, “Improving mention detection robustness to noisy input,” in *Proc. Conf. on Empirical Methods in Natural Language Process.*, Cambridge, MA, 2010, pp. 335–345.
- [54] I. Zitouni and R. Florian, “Mention detection crossing the language barrier.” in *Proc. Conf. on Empirical Methods in Natural Language Process.*, Honolulu, HI, 2008, pp. 600–609.
- [55] T. Ohta, S. Pyysalo, J. Tsujii, and S. Ananiadou, “Open-domain anatomical entity mention detection,” in *Proc. Workshop on Detecting Structure in Scholarly Discourse*, Jeju Island, Korea, 2012, pp. 27–36.
- [56] A. Sun, R. Grishman, and S. Sekine, “Semi-supervised relation extraction with large-scale word clustering,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Portland, OR, 2011, pp. 521–529.

- [57] A. Culotta and J. Sorensen, “Dependency tree kernels for relation extraction,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Barcelona, Spain, 2004, pp. 423–429.
- [58] L. Qian and G. Zhou, “Clustering-based stratified seed sampling for semi-supervised relation classification,” in *Proc. Conf. on Empirical Methods in Natural Language Process.*, Cambridge, MA, 2010, pp. 346–355.
- [59] L. Qian, G. Zhou, F. Kong, Q. Zhu, and P. Qian, “Exploiting constituent dependencies for tree kernel-based semantic relation extraction,” in *Proc. Int. Conf. on Computational Linguistics*, Manchester, UK, 2008, pp. 697–704.
- [60] B. Plank and A. Moschitti, “Embedding semantic similarity in tree kernels for domain adaptation of relation extraction,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Sofia, Bulgaria, 2013, pp. 1498–1507.
- [61] H. Ji and R. Grishman, “Improving name tagging by reference resolution and relation detection,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Ann Arbor, MI, 2005, pp. 411–418.
- [62] D. Roth and W. Yih, “A linear programming formulation for global inference in natural language tasks,” in *Proc. Conf. on Natural Language Learning*, Boston, MA, 2004, pp. 1–8.
- [63] D. Roth and W. Yih, “Global inference for entity and relation identification via a linear programming formulation,” in *Introduction to Statistical Relational Learning*. Cambridge, MA: MIT Press, Jan. 2007, ch. 20, pp. 553–580.
- [64] B. Yang and C. Cardie, “Joint inference for fine-grained opinion extraction,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Sofia, Bulgaria, 2013, pp. 1640–1649.
- [65] R. J. Kate and R. Mooney, “Joint entity and relation extraction using card-pyramid parsing,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Uppsala, Sweden, 2010, pp. 203–212.
- [66] P. Jindal and D. Roth, “Using soft constraints in joint inference for clinical concept recognition,” in *Proc. Conf. on Empirical Methods in Natural Language Process.*, Seattle, WA, 2013, pp. 1808–1814.
- [67] V. Punyakanok, D. Roth, W. Yih, and D. Zimak, “Learning and inference over constrained output,” in *Int. Joint Conf. on Artificial Intell.*, Edinburgh, UK, 2005, pp. 1124–1129.
- [68] M. Chang, L. Ratinov, and D. Roth, “Structured learning with constrained conditional models,” *Mach. Learning*, vol. 88, no. 3, pp. 399–431, Jun. 2012.

- [69] A. M. Rush, D. Sontag, M. Collins, and T. Jaakkola, “On dual decomposition and linear programming relaxations for natural language processing,” in *Proc. Conf. on Empirical Methods in Natural Language Process.*, Boston, MA, 2010, pp. 1–11.
- [70] C. A. Sutton, K. Rohanimanesh, and A. McCallum, “Dynamic conditional random fields: factorized probabilistic models for labeling and segmenting sequence data,” in *Proc. Int. Conf. on Mach. Learning*, Banff, Canada, 2004.
- [71] M. Wick, S. Singh, and A. McCallum, “A discriminative hierarchical model for fast coreference at large scale,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Jeju Island, Korea, 2012, pp. 379–388.
- [72] H. Poon and P. Domingos, “Joint inference in information extraction,” in *Proc. AAAI Conf. on Artificial Intell.*, Vancouver, Canada, 2007, pp. 913–918.
- [73] H. Poon and L. Vanderwende, “Joint inference for knowledge extraction from biomedical literature,” in *Proc. Conf. of the North Amer. Chapter of the Assoc. for Computational Linguistics*, Los Angeles, CA, 2010, pp. 813–821.
- [74] C. Kiddon and P. Domingos, “Knowledge extraction and joint inference using tractable markov logic,” in *Proc. Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, Montreal, Canada, 2012, pp. 79–83.
- [75] Y. Zhang and S. Clark, “Joint word segmentation and pos tagging using a single perceptron,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Columbus, OH, 2008, pp. 888–896.
- [76] X. Liu, M. Zhou, F. Wei, Z. Fu, and X. Zhou, “Joint inference of named entity recognition and normalization for tweets,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Jeju Island, Korea, 2012, pp. 526–535.
- [77] X. Qian and Y. Liu, “Joint chinese word segmentation, pos tagging and parsing,” in *Proc. Conf. on Empirical Methods in Natural Language Process.*, Jeju Island, Korea, 2012, pp. 501–511.
- [78] X. Zeng, D. Wong, L. Chao, and I. Trancoso, “Graph-based semi-supervised model for joint chinese word segmentation and part-of-speech tagging,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Sofia, Bulgaria, 2013, pp. 770–779.
- [79] J. R. Finkel and C. D. Manning, “Hierarchical joint learning: Improving joint parsing and named entity recognition with non-jointly labeled data,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Uppsala, Sweden, 2010, pp. 720–728.

- [80] X. Yu and W. Lam, “Jointly identifying entities and extracting relations in encyclopedia text via a graphical model approach,” in *Proc. Int. Conf. on Computational Linguistics*, Beijing, China, 2010, pp. 1399–1407.
- [81] M. Wang, W. Che, and C. D. Manning, “Joint word alignment and bilingual named entity recognition using dual decomposition,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Sofia, Bulgaria, 2013, pp. 1073–1082.
- [82] I. Meza-Ruiz and S. Riedel, “Jointly identifying predicates, arguments and senses using markov logic,” in *Proc. Conf. of the North Amer. Chapter of the Assoc. for Computational Linguistics*, Boulder, CO, 2009, pp. 155–163.
- [83] T. Zhuang and C. Zong, “Joint inference for bilingual semantic role labeling,” in *Proc. Conf. on Empirical Methods in Natural Language Process.*, Cambridge, MA, 2010, pp. 304–314.
- [84] P. Li, Q. Zhu, H. Diao, and G. Zhou, “Jointly modeling of trigger identification and event type determination in chinese event extraction,” in *Proc. Int. Conf. on Computational Linguistics*, Mumbai, India, 2012, pp. 1635–1652.
- [85] M. Wick, S. Singh, H. Pandya, and A. McCallum, “A joint model for discovering and linking entities,” in *Proc. 3rd Int. Workshop on Automated Knowledge Base Construction*, San Francisco, CA, 2013, pp. 67–72.
- [86] S. Singh, S. Riedel, B. Martin, J. Zheng, and A. McCallum, “Joint inference of entities, relations, and coreference,” in *Proc. 3rd Workshop on Automated Knowledge Base Construction*, San Francisco, CA, 2013.
- [87] M. Collins, “Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms,” in *Proc. Conf. on Empirical Methods in Natural Language Process.*, Philadelphia, PA, 2002, pp. 63–70.
- [88] M. Collins and B. Roark, “Incremental parsing with the perceptron algorithm,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Barcelona, Spain, 2004, pp. 111–118.
- [89] L. Huang and K. Sagae, “Dynamic programming for linear-time incremental parsing,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Uppsala, Sweden, 2010, pp. 1077–1086.
- [90] Y. Zhang and S. Clark, “A fast decoder for joint word segmentation and pos-tagging using a single discriminative model,” in *Proc. Conf. on Empirical Methods in Natural Language Process.*, Cambridge, MA, 2010, pp. 843–852.

- [91] J. Hatori, T. Matsuzaki, Y. Miyao, and J. Tsujii, “Incremental joint pos tagging and dependency parsing in chinese,” in *Proc. Int. Joint Conf. on Natural Language Process.*, Chiang Mai, Thailand, 2011, pp. 1216–1224.
- [92] H. Daumé III, “Practical structured learning techniques for natural language processing,” Ph.D. dissertation, Dept. Comput. Sci., Univ. Southern California, Los Angeles, CA, 2006.
- [93] Y. Kamide, G. Altmann, and S. L. Haywood., “The time-course of prediction in incremental sentence processing: Evidence from anticipatory eye movements,” *J. of Memory and Language*, vol. 49, no. 1, pp. 133–156, Jul. 2003.
- [94] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, “Conditional random fields: Probabilistic models for segmenting and labeling sequence data,” in *Proc. Int. Conf. on Mach. Learning*, Williamstown, MA, 2001, pp. 282–289.
- [95] A. K. McCallum, “Mallet: A machine learning for language toolkit,” 2002, unpublished. [Online]. Available: <http://mallet.cs.umass.edu> (Date Last Accessed, March, 10, 2015)
- [96] M.-C. D. Marneffe, B. Maccartney, and C. D. Manning, “Generating typed dependency parses from phrase structure parses,” in *Proc. Int. Conf. on Language Resources and Evaluation*, 2006, pp. 449–454.
- [97] C. Cortes and V. Vapnik, “Support-vector networks,” *Mach. Learning*, vol. 20, no. 3, pp. 273–297, Jul. 1995.
- [98] C. Macleod, R. Grishman, A. Meyers, L. Barrett, and R. Reeves, “Nomlex: A lexicon of nominalizations,” in *Proc. EURALEX Int. Congr.*, Liege, Belgium, 1998, pp. 187–193.
- [99] P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai, “Class-based n-gram models of natural language,” *Computational Linguistics*, vol. 18, no. 4, pp. 467–479, Dec. 1992.
- [100] R. McDonald, K. Crammer, and F. Pereira, “Online large-margin training of dependency parsers,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Ann Arbor, MI, 2005, pp. 91–98.
- [101] L. Huang, S. Fayong, and Y. Guo, “Structured perceptron with inexact search,” in *Proc. Conf. of the North Amer. Chapter of the Assoc. for Computational Linguistics*, Montreal, Canada, 2012, pp. 142–151.
- [102] Y. Freund and R. E. Schapire, “Large margin classification using the perceptron algorithm,” *Mach. Learning*, vol. 37, no. 3, pp. 277–296, Dec. 1999.

- [103] S. Sarawagi and W. W. Cohen, “Semi-markov conditional random fields for information extraction,” in *Proc. Annu. Conf. on Neural Inform. Process. Syst.*, Vancouver, Canada, 2004, pp. 1185–1192.
- [104] C. F. Baker and H. Sato, “The framenet data and software,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Sapporo, Japan, 2003, pp. 161–164.
- [105] M.-C. de Marneffe and C. D. Manning, *Stanford Dependencies Manual*, Stanford, CA, 2008. [Online]. Available: http://nlp.stanford.edu/software/dependencies_manual.pdf (Date Last Accessed, April, 23, 2015)
- [106] J. Dölling, “Aspectual coercion and eventuality structure,” in *Events, Arguments, and Aspects, Topics in the Semantics of Verbs*. Amsterdam, Netherlands: John Benjamins, Jun. 2014, ch. 2, pp. 189–226.
- [107] T. Van de Cruys, “Two multivariate generalizations of pointwise mutual information,” in *Proc. Workshop on Distributional Semantics and Compositionality*, Portland, OR, 2011, pp. 16–20.
- [108] K. Bollacker, R. Cook, and P. Tufts, “Freebase: A shared database of structured general human knowledge,” in *Proc. AAAI Conf. on Artificial Intell.*, Vancouver, Canada, 2007, pp. 1962–1963.
- [109] P.-C. Chang, M. Galley, and C. D. Manning, “Optimizing chinese word segmentation for machine translation performance,” in *Proc. 3rd Workshop on Statistical Mach. Translation*, Columbus, OH, June 2008, pp. 224–232.
- [110] C. Sutton, A. McCallum, and K. Rohanimanesh, “Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data,” *J. Mach. Learning Res.*, vol. 8, pp. 693–723, Mar. 2007.
- [111] M. J. Wainwright, T. Jaakkola, and A. S. Willsky, “Tree-based reparameterization for approximate inference on loopy graphs,” in *Proc. Annu. Conf. on Neural Inform. Process. Syst.*, 2001, pp. 1001–1008.
- [112] F. J. Och and H. Ney, “Improved statistical alignment models,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Hong Kong, China, 2000, pp. 440–447.
- [113] H. Ji, R. Grishman, D. Freitag, M. Blume, J. Wang, S. Khadivi, R. Zens, and H. Ney, “Name extraction and translation for distillation,” in *Handbook of Natural Language Processing and Machine Translation: DARPA Global Autonomous Language Exploitation*. New York, NY: Springer, Mar. 2011, ch. 4, pp. 617–716.

- [114] J. Zheng, “Srinterp: Sri’s scalable multipurpose smt engine,” SRI International, Menlo Park, CA, Tech. Rep., Jun. 2008. [Online]. Available: <http://www.speech.sri.com/projects/translation/srinterp.pdf> (Date Last Accessed, April, 23, 2015)
- [115] J. Zheng, N. F. Ayan, W. Wang, and D. Burkett, “Using Syntax in Large-Scale Audio Document Translation,” in *Proc. Annu. Conf. of Int. Speech Commun. Assoc.*, Brighton, UK, 2009, pp. 440–443.
- [116] D. Chiang, “A hierarchical phrase-based model for statistical machine translation,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Ann Arbor, MI, 2005, pp. 263–270.
- [117] F. J. Och and H. Ney, “A systematic comparison of various statistical alignment models,” *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, Mar. 2003.
- [118] K. Papineni, S. Roukos, T. Ward, and W. Zhu, “BLEU: a Method for Automatic Evaluation of Machine Translation,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Philadelphia, PA, 2002, pp. 311–318.
- [119] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, “A Study of Translation Edit Rate with Targeted Human Annotation,” in *Proc. Assoc. for Mach. Translation in the Americas*, Boston, MA, 2006, pp. 223–231.
- [120] X. Li, S. Strassel, S. Grimes, S. Ismael, X. Ma, N. Ge, A. Bies, N. Xue, and M. Maamouri, “Parallel aligned treebank corpora at ldc: Methodology, annotation and integration,” in *Proc. Workshop on Annotation and Exploitation of Parallel Corpora*, Tartu, Estonia, 2010.
- [121] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, “Distant supervision for relation extraction without labeled data,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Suntec, Singapore, 2009, pp. 1003–1011.
- [122] M. Bansal, D. Burkett, G. de Melo, and D. Klein, “Structured learning for taxonomy induction with belief propagation,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Baltimore, MA, 2014, pp. 1041–1051.
- [123] X. Zhu, “Semi-supervised learning literature survey,” Dept. Comput. Sci., Univ. Wisconsin-Madison, Madison, WI, Tech. Rep. 1530, Jun. 2005.
- [124] A. Rahman and V. Ng, “Coreference resolution with world knowledge,” in *Proc. Annu. Meeting of the Assoc. for Computational Linguistics*, Portland, OR, 2011, pp. 814–824.

- [125] L. Ratinov, “Exploiting knowledge in nlp,” Ph.D. dissertation, Dept. Comput. Sci., Univ. Illinois at Urbana-Champaign, Urbana, IL, 2012.