BACKGROUND KNOWLEDGE IN LEARNING-BASED RELATION EXTRACTION

BY

QUANG XUAN DO

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2012

Urbana, Illinois

Doctoral Committee:

Professor Dan Roth, Chair
Assistant Professor Julia Hockenmaier
Assistant Professor Heng Ji, City University of New York
Associate Professor ChengXiang Zhai

Abstract

In this thesis, we study the importance of background knowledge in relation extraction systems. We not only demonstrate the benefits of leveraging background knowledge to improve the systems' performance but also propose a principled framework that allows one to effectively incorporate knowledge into statistical machine learning models for relation extraction. Our work is motivated by the fact that relation extraction systems in the literature usually use evidence that is written explicitly in the input text to detect and characterize the semantic relations between target concepts. Although this approach achieves reasonable performance, it does not necessarily guarantee accurate extraction due to problems of poor information representation of the systems' inputs and lack of knowledge to support logical reasoning. We argue that relation extraction systems would benefit from using one or more background knowledge sources, both in enriching the systems' inputs and biasing the final outputs. We illustrate our framework in the context of several learning-based relation extraction tasks. The first task is *Taxonomic Relation Identification* where we employ an external knowledge source to construct meaning representation of the task inputs and support global inference to identify taxonomic relations between input terms. In the second task, *Event Relation Discovery*, we focus on identify causality relation between events in text. Our approach leverages background knowledge to perform joint inference among several classifiers that make local decisions on event causality relation. After that, we study the problem of constructing a timeline of events extracted from text, *Event Timeline Construction*. To address this task, we propose a new timeline representation with events mapped to absolute time intervals. In this work, we present a time interval-based global inference model that jointly assigns events into time intervals on a timeline and orders events temporally. Besides using relational constraints in the inference model, we also show that using event coreference as another source of background knowledge is beneficial to the system.

To My Family.

Acknowledgments

My thesis should have not been accomplished without the help of many individuals. My memory about the first days that I came to the University of Illinois at Urbana-Champaign and all people that contribute to my work is still fresh. Although I really want, I, however, realize that writing down the acknowledgement to all the individuals is frustratingly impossible. I would love to feature some of the contributions that others have had on my dissertation. This could by no means express my immense appreciation to all individuals that I am in debt for this work.

First and foremost, I would like to thank my parents, my brother Hoang Do and his wife, and my relatives (especially, the Vodinhs) for their huge support during my PhD life. Particularly, I want to express my deep gratitude to my mother, Loan Le, who has sacrificed so much for me so that I can be as I am today. Thank you, Mom. For all you have done for me, I found that writing down some acknowledgement sentences is impossibly sufficient. My gratefulness also goes to my dear wife, Nhung, for her love, support and encouragement. One of the best achievements that we have gone through together was the birth of our beloved daughter, Sophia Nhat-Phuong, just about two months before my final thesis defense. Sophia has made our life more joyful and impassioned.

I am deeply grateful to my advisor, Dan Roth, for his knowledgeable advices and supports from day one of my PhD journey. Dan was one of the most intellectually admirable individual that I have ever met and worked with. Dan was a tremendous advisor to me. He was always patient and supportive even when I was in long procrastinations. And of course, Dan has put huge influence on the improvement of my English skills. I will also always remember his great personalities in communicating and working with others. I believe that without his guidance and supports, my thesis would not have been realized. It was my true privilege to meet and work with you, Dan.

It was also my great honor to have other supportive members in my doctoral committee:

Julia Hockenmier, Heng Ji (from the City University of New York), and ChengXiang Zhai. Their academic work has inspired me much. Their insightful and thorough comments on my work indeed helped me improve my dissertation remarkably.

I was also lucky to be a member of the Cognitive Computation Group (led by Dan Roth) with several brilliant colleagues who supported me in research. My dissertation benefited from all individuals with whom I had chances to collaborate during my PhD journey, including YeeSeng Chan, Ming-Wei Chang, Wei Lu, Mark Sammons, Vivek Srikumar, Yuancheng Tu, Vinod Vydiswaran and Ran Zhao. In addition, my knowledge has been enlightened by exchanging and consulting ideas from many other members of the group: Kai-Wei Chang, James Clarke, Michael Connor, Dan Goldwasser, Prateek Jindal, Alex Klementiev, Gourab Kundu, Jack Morrissey, Jeff Pasternack, Vasin Punyakanok, Lev Ratinov, Nick Rizzolo, Alla Rozovskaya, Rajhans Samdani, and Kevin Small. On another perspective, all discussions and conversations that I had with my colleagues did implicitly help me improve my English skills. Particularly, I still remember many conversations with Jeff, which seemed to go nowhere, actually benefited my English skills substantially. I also enjoyed several table tennis matches and foosball games with Ming-wei, Kai-wei and Jeff.

Besides my work, I was fortunate to spend my life with many good friends and families at UIUC, especially the members of the Vietnamese Student Community, who contributed to keeping my life balanced. It is impractical to name all of them here, but I would like to highlight some individuals who were my roommates, exercising partners, drinking buddies and also work collaborators: Loc Bui, Thu Dang, Quang Dinh, Phong Le, Tung Le, Nam Nguyen, Tuan Hoang, Hoang Nguyen, Kien Nguyen, Minh Pham, Trong Tong, Duan Tran, Anh Truong, Loan Vo, and Long Vu.

Finally, I gratefully acknowledge the sponsorship of the Vietnam Education Foundation. The work in my dissertation was supported by the Defense Advanced Research Projects Agency, Machine Reading Program under Air Force Research Laboratory prime contract No. FA8750-09-C-0181, and the Army Research Laboratory under agreement W911NF-09-2-0053.

June, 2012.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In the age of information, people can be easily overwhelmed by the vast amount of information released every day. To help users access this information efficiently, we need better ways to automatically recognize and extract useful information from text with good and practical information extraction (IE) systems. For example, a system could regularly identify/summarize and then feed us recently published events of interest from the news.

Let us consider the following text snippet. We want to identify the events in the text (informally, we want to know who does what to whom, where and when).

> Iraq held elections on 3/07/2010, to elect a new Parliament and a prime minister. The slate led by Prime Minister Nuri Kamal al-Maliki trailed one led by a former interim leader, Ayad Allawi, by 89 seats to 91.

An IE system is expected to extract the events conveyed in the text, such as *holdsElections*(Prime Minister, Iraq, [Nuri Kamal al-Maliki, Ayad Allawi], 3/07/2010). The *holdsElection* event is defined by the following components:

- **Predicate:** *held elections*, is the language marker (a.k.a. event trigger) that informs the existence of an election event in the text.
- **Arguments:** *Prime Minister* is the position of the election, *Iraq* is the organization holding the event, and *Nuri Kamal al-Maliki* and *Ayad Allawi* are the candidates running for the election. Furthermore, 3/07/2010 describes the temporal information of the event.

The work of this thesis argues that in order to understand the text to parse the events at that level, there is a need to use information that is not in the text, which we call here background knowledge. Without the support from background knowledge, a system may possibly make mistakes

by identifying *Nuri Kamal al-Maliki* and *Ayad Allawi* as the positions or the organizations of the election. It would be very valuable if we can tell from the background knowledge that they are both *politicians*, thus, they should be more likely to be the election candidates. There has been much previous work addressing problems in relation extraction. However, most of the systems focus on extracting information explicitly expressed in the given text, despite the fact that implicit background knowledge is usually essential to supporting precise extraction.

1.1 Thesis Contributions

In this thesis, we argue that background knowledge largely helps relation extraction systems to improve performance. We propose a principled framework that incorporate multiple background knowledge sources into extraction models to precisely and comprehensively recognize and extract relations from text. This thesis studies three important instances of the relation extraction problems that illustrate different aspects of information extraction and different approaches to exploit the use of background knowledge.

- **Taxonomic Relation Identification:** A lexicon-oriented problem, where inputs are well-segmented terms and outputs are their taxonomic relations.
- **Event Relation Discovery:** This problem focuses on discovering causality relations between events in free text.
- **Reasoning on Event Temporal Relations:** We address the problem of recognizing temporal relations between events with an emphasis on constructing a timeline of events from text.

As we discussed in the event extraction example above, it is important to recognize that both *Nuri Kamal al-Maliki* and *Ayad Allawi* are politicians. In other words, it is essential to identify that *politician* holds an ancestor relation (i.e. subsumes on a taxonomy) to *Nuri Kamal al-Maliki*, and also to *Ayad Allawi*. We address this problem as a taxonomic relation identification problem [Do and Roth, 2010, Do and Roth, 2012a], where we identify the taxonomic relations between any two given terms. We focus on recognizing *ancestor/descendant* and *sibling* relations. For example,

the term *the US president* was identify as an ancestor of the term *Barack Obama*, whereas the term *Apple Inc.* is a sibling of the term *Microsoft*. In our work, we use a machine learning approach where an external knowledge source (e.g. Wikipedia) is extensively used to support identifying taxonomic relations. We demonstrate that background knowledge has significant contributions to this task.

For the problem of discovering event relations, we address the task of extracting causality relations between events in free text. For this problem, we not only propose a new metric to measure the association strength of event causality, but also demonstrate the necessity of discourse constraints to further improve extraction quality [Do et al., 2011].

The third problem that we address is to automatically order events that happened in a news story on an absolute timeline represented by a set of time intervals extracted from the news. Event timeline will allow us to present events and their arguments in a meaningful representation over time, which will be highly useful to satisfy users' needs of daily information update. Specifically, we assign events to their corresponding time intervals and determine the partial temporal order among the events. In this work, we first introduce our robust and shallow temporal reasoning system that performs temporal expression extraction, normalization to time intervals and comparison [Zhao et al., 2012]. Next, we propose a joint timeline model that couples the decisions of local classifiers with respect to a set of relational constraints to improve the quality of constructing a timeline of events. This work also discusses the importance of event coreference as a background knowledge source in supporting the task at hand [Do and Roth, 2012b].

For a short summary, the most significant contribution of this thesis is proposing advanced and expressive models that combine statistical machine learning approach with background knowledge to address relation extraction tasks. To this end, this research present a principled framework and develop approaches to represent and incorporate external knowledge and computational constraints into relation extraction systems to achieve better performance for relation extraction systems. This work helps improve information extraction applications in order to better serve the real-world needs of people in the age of information.

1.2 The Use of Background Knowledge in Relation Extraction

In this section, we categorize background knowledge sources and discuss their use in relation extraction tasks.

We characterize background knowledge sources by four types as follows:

- *Facts:* the knowledge that states facts in the world. For example, *Chicago* locates in *Illinois*, and *Ernest Hemingway* is the author of *The Old Man and the Sea*.

- *Common Sense:* the information that can be ordinarily observed in the world. An example of common sense knowledge is transitivity closure, such as given that *President Bill Clinton* precedes *President George Bush* who precedes *President Barack Obama*, one can infer that *President Bill Clinton* precedes *President Barack Obama*.

- *Distributional knowledge:* the knowledge that is inferred from large text collections. This type of background knowledge is also referred to as *distributional similarity* and *distributional semantic*. Due to the nature of this kind of knowledge, it is common to have a confidence probability associated with each piece of knowledge. For example, by counting from a huge document collection, one observes that 90% of time *Bill* is a nickname of *William*, while *Tom* is a nickname of *Thomas* with 75% of confidence.

- *Linguistics:* the knowledge from linguistics. For example, the type of discourse relation between two text spans can tell us if there is a causality relation between the main verbs in the text spans. In the following sentence: *The first priority is search and rescue because many people are trapped under the rubble.*, the verb *trapped* signals a cause-effect relation on the verbs *search* and *rescue* via the discourse connective *because*. The specific linguistic knowledge of this example is that a *Contingency* discourse relation between two text spans indicates that the situation described in one text span causally influences the situation in the other.

In this work, we show that background knowledge can be used flexibly to make influence on different aspects of relation extraction tasks. We make use of the four types of background knowledge both in (i) elaborating tasks' inputs and intermediate learning features, and (ii) biasing

final outputs. We exercise the use of background knowledge in the three aforementioned relation extraction tasks and show that background knowledge positively contributes to the performance gains of the systems. Specifically, we employ background knowledge in the tasks as follows:

- *Taxonomic relation identification:* In this task, we make use of an external knowledge source (Wikipedia) that provides *facts* to enrich inputs and employ *common sense* to enforce global consistency on outputs. Furthermore, we also use *distributional semantic* knowledge learned from Wikipedia to construct learning features for a local classifier.

- *Event relation discovery:* The main contribution of this work is inventing a new metric that measures the cause-effect association strength between any two events in a text. This metric relies heavily on *distributional semantic* knowledge derived from a large document collection. We further improve the system performance by incorporating *linguistics* knowledge on discourse relations in text to capture global coherence in event causality recognition as the output of the system.

- *Reasoning on event temporal relations:* Our work on temporal reasoning focuses on associating events to their corresponding time intervals in text and building temporal partial order among the events. We illustrate the important role of *common sense* knowledge in solving the task by providing important relational constraints enforcing a global agreement among local classifiers. Furthermore, *linguistics* knowledge on event coreference is also shown to be very beneficial in biasing the final outputs of the system.

Next, we will discuss the approaches that allow us to integrate background knowledge into the systems, based on their uses.

- *Enriching tasks' inputs:* Technically, background knowledge is used to map tasks' inputs to a more informative input space. The intuition is that background knowledge can provide more information on the inputs so that core extraction algorithms can do more reasoning on the inputs to achieve better performance on extracting target relations. For example, in the task of taxonomic relation extraction, the system is required to recognize the taxonomic relation between two terms such as (*actor*, *Russell Crowe*). It is possible that one can directly infer

the relation between the input terms if the relation of the terms has been learned in advance. However, if the terms and their relation have not been seen before, one still has to identify the relation between two terms *actor* and *Russell Crowe*, which provide very little information about themselves. In this case, we use background knowledge to first enrich the input terms and then apply the core identification algorithm on the new input space. Using *facts* and *distributional knowledge*, we can elaborate the terms as follows: *actor* is mapped to a new bag of words *actor, actress, person, dramatic, comic, production, film, movie, television, theatre, radio, character, scene, ...*, and *Russell Crowe* is mapped to *russell, crowe, australian, actor, film, producer, musician, ...* It is clear that we now have richer information about the input terms so that we can identify their taxonomic relation more accurately.

- *Biasing final outputs:* On this perspective, background knowledge is used to enforce global consistency among predicted outputs. Intuitively, the outputs in relation extraction tasks usually interact with each other to maintain a global coherent structure of relations. Background knowledge provides global constraints that can be integrated into extraction systems to make influences on the final predictions. For example, in the task of taxonomic relation identification, by using *common sense knowledge* on transitivity closure, we require a system that predicts *Beethoven* as a taxonomical sibling of both *Mozart* and *Chopin* to recognize sibling as the taxonomic relation between *Mozart* and *Chopin*.

Our principled framework combines these two ways of using background knowledge into a pipeline that allows one to integrate multiple knowledge sources into relation extraction models. We show that background knowledge that is used in our proposed framework does significantly improve the performance of the systems.

1.3 Thesis Organization

The rest of the thesis is organized as follows:

- Chapter 2 provides a brief overview of the background in relation extraction and approaches to incorporating background knowledge into relation extraction tasks.

- Chapter 3 introduces the first use case where background knowledge is employed to support the task of taxonomic relation identification. In this work, we present the idea of using Wikipedia to enrich the feature space of input terms in order to effectively identify the taxonomic relation between them. Moreover, we propose relational constraints that enforce global coherence among local decisions to further improve the quality of our system.

- Chapter 4 describes our work on the event relation discovery task. In this work, we focus on extracting events in news text and retrieving causality relation between pair of events. Besides proposing a new metric to event causality discovering, we show that incorporating discourse relations as background knowledge improves the system performance significantly.

- Chapter 5 addresses the problem of building a temporal reasoning system that can automatically order event into an absolute timeline. We demonstrate that leveraging relational constraints as background knowledge in a joint inference model remarkably improves the quality of the constructed timeline. Moreover, we present event coreference as another source of background knowledge that is also beneficial for the final system.

- Last but not least, chapter 6 summarizes and concludes the work in this thesis.

Chapter 2

Background

In this section, we review previous work on relation extraction and background knowledge integration. We first review typical approaches that address the problem of extracting interesting relations from text. After that, we examine recent efforts in incorporating background knowledge into extraction processes.

2.1 Relation Extraction

Relation extraction is a field in natural language processing that focuses on detecting and characterizing the semantic relations of interest between target concepts in text [Jiang, 2012]. The role of relation extraction is more and more important because of the existence of a vast amount of unstructured data (under the form of text). Needless to say, it will be largely beneficial for human in understanding and acquiring information if relation extraction systems are realized. Intuitively, the systems will help turning unstructured data into structured knowledge bases with a focus on relations between target concepts such as entities or events. For example, in an entity relation extraction task, a system is expected to recognize the relation between entities, which can be person, location, organization, and so on. Examples of target relations include *locate_at(company, city)*, *author(writer, book)*, *affiliate(person, organization)*, etc. as in *locate_at(Facebook, Menlo Park)*, *author(Dan Brown, The Da Vinci Code)*, *affiliate(Larry Ellison, Oracle)*. For relations between events, some examples include: *bombing event* causes a *demonstration event*, and the *bombing event* precedes the *demonstration event* in time. We group relation extraction systems into two main groups based on their algorithmic approaches: supervised and unsupervised.

Work in the first group formulates the tasks as supervised classification problems: given a pair of target concepts, the systems classify and return the relation between the concepts. There

has been much work following this approach such as [Zhou et al., 2005, Jiang and Zhai, 2007, Chan and Roth, 2010]. In this approach, the systems either extract learning features or define kernels to classify the relation of input concepts. Usually, using correct features will produce very good performance in a supervised system. However, selecting good features is not a simple task. Some features are informative and useful to recognize correct relations while others are not, so there is a difficulty on deciding which features to use in a classifier. In order to overcome the problem of selecting good learning features, kernels are employed. Kernels can be defined at several aspects and levels such as subsequence kernels [Bunescu and Mooney, 2006], dependency trees and paths kernels [Culotta and Sorensen, 2004, Bunescu and Mooney, 2005], and also full parse tree kernels [Zelenko et al., 2003, Zhang et al., 2006]. Although the supervised approach provides the best performance in relation extraction, its largest hurdles are lack of annotated data and scalability. The research community of the field recently devotes a large amount of effort into developing algorithmic approaches that make use of no annotated data (unsupervised) or very little annotation (weakly-supervised).

In the second group, unsupervised/weakly-supervised methods are employed. Relation extraction bootstrapping algorithms, such as the work in [Hearst, 1992b, Pantel and Pennacchiotti, 2006, Kozareva et al., 2008], automatically harvest related terms on large corpora by starting with a few seeds of pre-specified relations (e.g. *is-a*, *part-of*). Bootstrapping algorithms rely on a scoring function to assess the quality of terms and additional patterns extracted during bootstrapping iterations. Beside the problem of low extraction recall, another well-known limitation of these algorithms is that extracted related terms have to appear in a close proximity in text. For example, a classical extraction pattern used in [Hearst, 1992b] to extract hyponyms of a term is defined as follows: NP_0 *such as* $NP_1, NP_2, \cdots,$ *(and | or)* NP_n. In this pattern, $NP_1, NP_2, \cdots, NP_n$ are extracted as the hyponyms of NP_0. Obviously, this kind of lexico-syntactic patterns cannot extract hyponyms that are far away from NP_0 in text. Another issue of classical relation extraction algorithms is that they require the target relations to be pre-specified. This is problematic because if we move to extracting information in huge corpora, such as the Web, we may want to greedily harvest all the possible relations and their associated entities. Motivated by this intuition, [Banko and Etzioni, 2008, Davidov and Rappoport, 2008a] introduced Open Information

Extraction (OpenIE), a framework that can extract related terms from massive corpora without pre-specifying a list of relations. However, it can be argued that the advantage of OpenIE is also a disadvantage because for too many relations extracted; they are unfocused and not very useful for end-systems. Moreover, the relations extracted from huge data usually suffer from noise due to incoherent extractions. There has been much work in literature attempting to improve the OpenIE framework either by revising the framework itself or refining and digging out useful information from the output of OpenIE models [Ritter et al., 2008, Fader et al., 2011].

Along this direction, distributional semantic models (DSMs) recently have got more attention from the community. These models leverage distributional contexts where words appear to harvest words with their attributes and build up word representation in semantic vector space [Padó and Lapata, 2007, Turney and Pantel, 2010, Baroni and Lenci, 2010]. A general framework of DSMs which were described in [Baroni and Lenci, 2010] extracts significant contexts of given terms from large corpora. Consequently, a term can be represented by a vector of contexts in which it frequently appears. Any vector space model could then be used to cluster terms into semantic classes by measuring semantic similarity between their vectors of representation. Our work on the task of taxonomic relation identification (see Chapter 3) involves an experiment that compares our propose algorithmic approach with this type of knowledge.

Recently, researchers are not only focusing on extracting lexicon-oriented relations, such as taxonomic or part-of relations, but also addressing more complicated kinds of relations. One of the relation types that recently attracts more and more effort in the research community is the temporal order of events in unstructured texts. Part of this effort falls into the work of extracting causal relation between events because it is obvious that events can be partially ordered by causality relations. For example, a *bombing* event causes a *damaging* event, thus *bombing* must happen before *damaging*. Although prior work in event causality extraction in context is relatively sparse, there is much prior work concerning other semantic aspects of event extraction. [Ji and Grishman, 2008] extracts event mentions (belonging to a predefined list of target event types) and their associated arguments. [Chambers and Jurafsky, 2008b, Chambers and Jurafsky, 2009] chain events sharing a common (protagonist) participant. They define events as verbs and given an existing chain of events, they predict the next likely event involving the protagonist. [Riaz and Girju, 2010] directly

focus on the problem of event causality extraction by proposing the *Effect Control Dependency* metric to assert and build a data set of causal verb-headed text spans. We continue discuss our study on this task in Chapter 4. On the other hand, research in temporal reasoning tasks, which include detecting, extracting, normalizing and reasoning on temporal expressions and events in text, now get more attention from the community. A more detailed review on the related work of this research direction is presented in Chapter 5 (see Section 5.2).

2.2 Incorporating Background Knowledge with Joint Inference Models

There has been work in the literature proposing approaches to integrate background knowledge into relation extraction models. In the last some years, the community has turn attention to a successful framework that allows direct knowledge injection in extraction tasks: Integer Linear Programming (ILP) for natural language processing [Roth and Yih, 2007, Clarke and Lapata, 2008, Martins et al., 2009]. By using the ILP framework, one can perform global and/or joint inferences on the outputs of a system to enforce the global consistency among the outputs. An ILP inference model is usually equipped with a list of hard or soft constraints (enforced on the system outputs) to require the outputs to agree with each other, if they are overlap. For example, a typical constraint could be: *born_in* is a binary relation between two entities, the first entity must be a *person* and the second one must be a *location*. The ILP framework derives exact solutions for problems at hand.

Intuitively, the key idea of the ILP framework is maximizing an objective function, which is a linear combination of some local classifiers), while satisfying some restrictions imposed by constraints. An good example of this approach can be found in [Roth and Yih, 2007]. Recently, [Chang et al., 2007] propose a superset of machine learning linear models called the Constrained Conditional Model (CCM). CCM uses ILP as one of possible optimization models to derive exact solution to input problem.

Definition (CCM): *A CCM can be represented by two weight vectors, w and ρ, given a set of feature functions $\{\phi_i(.)\}$ and a small set of constraints $\{C_i(.)\}$. The score for an assignment $y \in \mathcal{Y}$ on an instance $x \in \mathcal{X}$ can then be obtained by*

$$f_C(x, y) = \sum w_i \phi_i(x, y) - \sum \rho_i C_i(x, y), \tag{2.1}$$

where each $C_i : \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}$ *is a Boolean function indicating whether the joint assignment* (x, y) *violates i-th constraint. A CCM then selects* $y^* = argmax_y f_C(x; y)$ *as its prediction.*

It is worth noting that if no constraint is given, CCM is exactly the same as other linear models in learning. Furthermore, it is possible to set the value of $\{\rho_i\}$ by using external knowledge. Usually, ρ is set to ∞ if we are confident about the knowledge. If the knowledge is not perfect, we can set ρ to some positive value.

The authors argue that constraints can be much more expressive than the features used by linear models. This can be explained by the fact that constraints have effects directly on the output space of a problem. They, therefore, can correct the outputs by overwriting the predictions produced by some local linear models. Furthermore, by using constraints, one can easily enforce a global consistency among the outputs. The constraints support one to maintain a highly consistent structure of the local prediction outputs.

In addition, CCM provides a framework that simplifies the model of a complex structured output problem. The key idea is that CCM allows one to perform joint inferences with the combination of several local learned models that respect to a set of constraints. On the other hand, a complex model may have to discover and extract complex features to be able to achieve a good performance.

In this thesis, we use CCM as a main inference framework to integrate background knowledge into relation extraction tasks.[1]

[1]We note that CCM is only a tool that we use to incorporate background knowledge into our proposed framework of using background knowledge to support relation extraction.

Chapter 3

Taxonomic Relation Identification

Determining whether two terms have an ancestor relation (e.g. *Toyota Camry* and *car*) or a sibling relation (e.g. *Toyota* and *Honda*) is an essential component of textual inference in Natural Language Processing applications such as Question Answering, Summarization, and Textual Entailment. Significant work has been done on developing knowledge sources that could support these tasks, but these resources usually suffer from low coverage, noise, and are inflexible when dealing with ambiguous and general terms, that may not appear in any stationary resource, making their use as general purpose background knowledge resources difficult. In this work, rather than building a hierarchical structure of concepts and relations, we describe an algorithmic approach that, given two terms, determines the taxonomic relation between them using a machine learning-based approach that makes use of existing resources. Moreover, we develop a global constraint-based inference process that leverages an existing knowledge base to enforce relational constraints among terms and thus improves the classifier predictions. Our experimental evaluation shows that our approach significantly outperforms other systems built upon existing well-known knowledge sources.

3.1 Introduction

Fundamental taxonomic relations such as *ancestor-descendant* (e.g. *actor* and *Mel Gibson*) and *siblings* (e.g. *Mel Gibson* and *Tom Cruise*) have been shown to hold important roles in many computational linguistics tasks, such as document clustering [Hotho et al., 2003], navigating text databases [Chakrabarti et al., 1997], Question Answering (QA) [Saxena et al., 2007] and Summarization [Vikas et al., 2008]. Recently, it has been shown that recognition of taxonomic relations between terms is essential to support textual inference tasks such as Textual Entailment (TE) [Dagan et al., 2006]. For example, it may be important to know that a *blue Toyota Prius* is nei-

ther a *white Toyota Prius* nor a *blue Toyota Camry*, and that all are *cars*. Work in TE has argued quite convincingly [MacCartney and Manning, 2008] that many such textual inferences are largely compositional and depend on the ability to recognize fundamental taxonomic relations, such as the ancestor or sibling relations, between terms. Furthermore, several TE studies [Abad et al., 2010, Sammons et al., 2010] suggest isolating TE phenomena, including recognizing taxonomic relations, and studying them separately. They also discuss characteristics of several phenomena (e.g. contradiction) from a perspective similar to ours, but do not provide a solution.

Motivated by the needs of natural language processing tasks, and the compositionality argument alluded to above, this chapter addresses the problem of classifying fundamental taxonomic relations between terms: given two well-segmented terms, the system predicts the taxonomic relation between them – *ancestor-descendant*, *siblings* or *no relation*. In this work, the context where the terms come from is not given. We leave the idea of leveraging the context of the input terms and how to use the taxonomic relations in applications to a future extension of this work.

An input term could be any well-segmented span of words that refers to a concept. Moreover, input terms may include common nouns or proper nouns from open or closed concept classes. Some examples of input terms include *mountain*, *George W. Bush*, *battle of Normandy*, *table*, *US Today*, *NATO*, and *chemical elements*. In this work, we use *term* and *concept* interchangeably, even though *concept* is usually used to refer to nodes in hierarchical resources. For taxonomic relations, we consider that two terms hold an *ancestor-descendant* relation if one term is subsumed by the other w.r.t. a taxonomic structure, whereas two terms are *siblings* if they share a common subsumer.

An ancestor-descendant relation and its directionality can help us infer that a text snippet mentioning a descendant term (e.g. *cannabis*) entails a hypothesis mentioning an ancestor term (e.g. *drugs*) in a similar way as in the following example, taken from a TE challenge data set.

> **Text**: Nigeria's NDLEA has seized 80 metric tons of *cannabis* in one of its largest ever hauls, officials say.
>
> **Hypothesis**: Nigeria seizes 80 tons of *drugs*.

Similarly, it is important to know of a sibling relation to infer that a statement about *Taiwan*

(without additional information) is not likely to entail a hypothesis about *Japan* since they are different countries, as in the following example:

> **Text**: A strong earthquake struck off the southern tip of *Taiwan* at 12:26 UTC, triggering a warning from Japan's Meteorological Agency that a 3.3 foot tsunami could be heading towards Basco, in the Philippines.
>
> **Hypothesis**: An earthquake struck *Japan*.

Naturally, these taxonomic relations can be read off from manually generated resources such as Wordnet that explicitly represent these relations. However, it is clear that these resources have limited coverage. For example, Wordnet 3.0 [Fellbaum, 1998] consists of only around 118,000 nominal concepts, which is obviously much smaller than the number of concepts in English. In addition, very few entities and multiword concepts are covered in WordNet.

There has also been work on extending the manually built resources using automatic acquisition methods resulting in structured knowledge bases such as the Extended WordNet [Snow et al., 2006] and the YAGO ontology [Suchanek et al., 2007]. These knowledge sources only partially alleviate the coverage problem, and could be potentially impaired by noise introduced when they were compiled.

One of the well-known approaches to building offline resources is using relational patterns (e.g. *X such as Y, Z*) to extract related terms from text [Hearst, 1992a, Snow et al., 2006]. Unfortunately, this approach is usually brittle. Infrequent terms are less likely to be covered, and may not be effectively extracted since they do not usually appear in close proximity with other terms (e.g. Israeli tennis player *Dudi Sela* and Swiss tennis champion *Roger Federrer* rarely appear together in news text). On the other hand, knowledge sources derived by using bootstrapping algorithms and distributional semantic models [Pantel and Pennacchiotti, 2006, Kozareva et al., 2008, Baroni and Lenci, 2010] typically suffer from a trade-off between precision and recall, resulting either in a relatively accurate resource with low coverage or a noisy resource with broader coverage.

Another limitation of structured resources, as we observe, is their inflexibility in dealing with terms that cannot be exactly mapped to existing concepts in the resources. This problem usually occurs when a resource actually contains a concept corresponding to an input term, but the concept

and the term are written with different surface strings. For example, one may not be able to map the input term *Chelsea* to concept *Chelsea, London* (an area of West London) in the Extended WordNet using an exact string-matching operation because their surface strings are not the same. Even worse, if the Extended Wordnet also maintains the concept *Chelsea F.C.* (an English football club based in West London) in addition to *Chelsea, London*, then there is no clear mechanism to map the input term *Chelsea* to the concept *Chelsea, London* or *Chelsea F.C.*[1]

In this chapter, we present a novel approach to identifying the taxonomic relation between two input terms by exploiting the rich structure and information of Wikipedia,[2] a free and collaboratively updated encyclopedia of concepts. It is important to emphasize that our work focuses on directly classifying relations that hold between input terms rather than building a resource of relational information among concepts. In this respect, we are distinct from Open Information Extraction [Banko et al., 2007], on-demand Information Extraction [Sekine, 2006], and other efforts to recognize facts in a given corpus [Davidov and Rappoport, 2008b, Paşca and Van Durme, 2008], which capitalize on local co-occurrence of terms to generate databases of open-ended facts. Our work is also different from the supervised relation extraction effort [Roth and Yih, 2004] that requires full text or sentences, where the two terms appear, to infer their relation.

In our work, we use Wikipedia as a background knowledge source. This resource has been shown to be very useful and powerful for many tasks in knowledge extraction such as in the work of [Suchanek et al., 2007, Ponzetto and Strube, 2007], information retrieval [Milne and Witten, 2008, Mihalcea and Csomai, 2007, Ratinov et al., 2011], and computing semantic relatedness such as in [Gabrilovich and Markovitch, 2007]. One of the most important advantages of Wikipedia is that it allows volunteers to contribute their knowledge collaboratively. Wikipedia, therefore, keeps growing over time with millions of relations and concepts, including common nouns and proper nouns (e.g. *chicken*, *blue*, *Everest*, *US Today*) and open and closed concept classes (e.g. *country*, *foods*, *chemical elements*). Specifically, Wikipedia was chosen for our work for the following reasons:

- Wikipedia consists of millions of pages providing rich information about concepts. The pages in Wikipedia are well organized in an informative structure. This allows us to easily leverage

[1] One may write a better coreference/ambiguity resolver to deal with ambiguous terms. However, it is not feasible when the context of the input terms is not given as in this work.

[2] http://wikipedia.org/

the information in Wikipedia to support classification decisions.

- The information in Wikipedia is collaboratively generated, modified and updated. Volunteers around the world contribute to Wikipedia everyday, guaranteeing that Wikipedia is up to date with new concept pages and improving old concept pages over time. Using Wikipedia as the background knowledge is semi-dynamic in the sense that Wikipedia is continuously growing and we can easily use the latest Wikipedia version into our classification framework.

- Wikipedia provides a complex system of redirect and disambiguation pages, which could be leveraged to overcome the problems of limited coverage and lack of surface matching.

- Each content page in Wikipedia contains, in addition to the concept description, also semantic categories of the concept. We take advantage of both the text and the categories in supporting taxonomic relation classification.

Our algorithmic approach takes two well-segmented terms as input and outputs the predicted taxonomic relation between them, focusing on *ancestor-descendant* and *sibling* relations. We first exploit the Wikipedia structure to build semantic representation of each input term. Next, learning features are extracted from the semantic representations of the terms. A learned multi-class classifier is then applied to the extracted features to predict a probability distribution over the relations. In addition, we present an inference model that makes use of relational constraints and the aforementioned probability distribution over the taxonomic relations of the two input terms and *additional related terms* to enforce a coherent structure of terms and predicted relations that support the final taxonomic relation prediction.

In the rest of this chapter, we present the overview of our algorithmic approach in Section 3.2. The learning component and the inference model of our approach are described in Sections 3.3 and 3.4. Experimental results showing the advantages of our system are described in Section 3.5. We briefly discuss related work in Section 3.6, and summarize the chapter in Section 3.7.

Label	Relation	Examples	
		Term x	**Term** y
$x \leftarrow y$	x is an **ancestor** of y	actor	Mel Gibson
		food	rice
		wine	Champagne
$x \rightarrow y$	x is a **descendant** of y	Makalu	mountain
		Monopoly	game
		krooni	currency
$x \leftrightarrow y$	x and y are **siblings**	Paris	London
		copper	oxygen
		London	Hemingway
$x \nleftrightarrow y$	x and y have **no relation**	Roja	C++
		egg	Vega
		HotBot	autism

Table 3.1: Four taxonomic relations and some examples of each relation. Note that *London* is an ambiguous concept. It can be a city, thus a sibling of *Paris*, but can also refer to *Jack London*, thus a sibling of *Hemingway*

.

3.2 Algorithmic Approach

3.2.1 Preliminaries

The basic problem that we address in this work is the identification of fundamental taxonomic relations between any two well-segmented terms. Instead of building structured resources that record taxonomic relations among concepts as in previous work, our system focuses on directly classifying any two input terms into fundamental taxonomic relations including *ancestor-descendant*, *siblings* or *no relation*.

The main component of our system is a taxonomic relation classifier that is trained on annotated data consisting of pairs of terms and their taxonomic relations. In order to directly identify the directionality of the relations between input terms, we explicitly train and evaluate the classifier on four relation labels — *ancestor*, *descendant*, *sibling* and *no relation*. Some examples in the training data, which consists of pairs of terms with four labels, are shown in Table 3.1.

It is worth noting that it is a pragmatic decision to determine whether two terms hold a taxonomic relation. For example, according to the Wikipedia category system, *George W. Bush* is a descendant of *Presidents of the United States* and also a descendant of *people*, *mammals*, and *organisms*. Without any constraint, the term *George W. Bush*, therefore, could be considered as

TAREC (Training)

INPUT	Annotated data $\mathcal{D} = \{(x, y, rel)\}$
	Wikipedia $\mathcal{W}$

ALGORITHM:

1.	$\mathcal{D}' = \emptyset$
2.	For each $(x, y, rel) \in \mathcal{D}$
3.	$\quad \mathfrak{R}_x \leftarrow \text{WikiRepresentation}(x, \mathcal{W})$
4.	$\quad \mathfrak{R}_y \leftarrow \text{WikiRepresentation}(y, \mathcal{W})$
5.	$\quad \mathcal{D}' = \mathcal{D}' \cup (\mathfrak{R}_x, \mathfrak{R}_y, rel)$
6.	$\mathcal{C} \leftarrow \text{ExtractFeaturesAndTrainClasifier}(\mathcal{D}')$
7.	RETURN: $\mathcal{C}$

Figure 3.1: The TAREC training algorithm.

a sibling of the term *oak (tree)* because they share *organisms* as a common subsumer. Obviously, we do not want to predict that *George W. Bush* and *oak* are siblings. In this work, we make use of Wikipedia structure as a source of background knowledge and use it to infer taxonomic relations between terms. Our taxonomic relation identifier uses a constant K – the maximum level to recursively climb up the Wikipedia category structure from a given concept – as a way to control determining of taxonomic relations between terms. Note that K is fixed for all relations and concepts

3.2.2 Overview

In this section, we present the overview of our **TA**xonomic **RE**lation **C**lassification (**TAREC**) system. The system consists of a training and an evaluation algorithms. Briefly, the training algorithm learns from a supervised training data set a local classifier that is used evaluation time in a constraint-based inference model to make the final prediction. We describe the algorithms below.

TAREC Training Algorithm

The training algorithm of TAREC is shown in Fig. 3.1. The input to the algorithm includes supervised training data $\mathcal{D}$ and Wikipedia data $\mathcal{W}$. The training data consists of examples in the form of triples (x, y, rel), where x and y are two terms and rel is their taxonomic relation. The relation rel denotes the taxonomic relation from x to y. For example, triple (*newspaper*, *New York*

TAREC (Evaluation)

INPUT	A pair of terms (x, y)
	Wikipedia $\mathcal{W}$
	Taxonomic relation classifier $\mathcal{C}$

ALGORITHM:

1. $\mathfrak{R}_x \leftarrow \text{WikiRepresentation}(x, \mathcal{W})$
2. $\mathfrak{R}_y \leftarrow \text{WikiRepresentation}(y, \mathcal{W})$
3. $\mathcal{P}_{x,y} \leftarrow \text{Classify}(\mathfrak{R}_x, \mathfrak{R}_y, \mathcal{C})$
4. $\mathcal{Z}_{x,y} \leftarrow \text{ExtractRelatedTerms}(x, y)$
5. $rel \leftarrow \text{ConstraintBasedInference}(\mathcal{P}_{x,y}, \mathcal{Z}_{x,y}, \mathcal{C})$
6. RETURN: rel

Figure 3.2: The TAREC evaluation algorithm.

Times, $\leftarrow$) denotes that *newspaper* is an ancestor of *New York Times*, while (*Canada*, *country*, $\rightarrow$) denotes that *Canada* is a descendant of *country*. Wikipedia data $\mathcal{W}$ is a local database constructed to allow access to necessary information in Wikipedia. We will discuss this background knowledge source in more details in Section 3.3.

To identify taxonomic relations between two single terms, we first map the terms to some informative representations from which we could extract useful features. The function WikiRepresentation(*term*, $\mathcal{W}$) constructs a Wikipedia-based semantic representation for the input *term*. A new learning example is formed from the Wikipedia representation of the two input terms and their gold taxonomic relation. The new data is then used to train a local multi-class classifier ($\mathcal{C}$) to predict relations. Note that beside predicting relations, the learned classifier can also predict relation directionality due to the fact that we explicitly have four relation labels in the training data — x is an ancestor of y, x is a descendant of y, x and y are siblings, and x and y have no relation. We consider the classifier returned from the TAREC training algorithm as a *local* classifier to distinguish it from the global inference process employed in the TAREC evaluation algorithm.

TAREC Evaluation Algorithm

Given two terms (x, y), we apply the TAREC evaluation algorithm to predict their taxonomic relation. The evaluation algorithm uses the local classifier $\mathcal{C}$ learned using the TAREC training algorithm to predict the probability distribution over four taxonomic relation labels of (x, y) with background knowledge source $\mathcal{W}$. As we do when training, the two input terms are first mapped to

a Wikipedia-based representation. The representations of x and y are then classified by $\mathcal{C}$ to get the probability distribution, $\mathcal{P}_{x,y}$, over the relation classes. Following that, the predicted probability distribution is used in a relational constraint-based inference model that takes advantage of other related concepts, $\mathcal{Z}_{x,y}$, of (x, y) to enforce a final coherent prediction on the taxonomic relation between x and y. In the inference model, we present a novel approach that leverages related concepts of two input terms, extracted from an existing knowledge source, to form a coherent relational structure that supports an accurate global prediction of taxonomic relations between input terms. The TAREC evaluation algorithm is summarized in Fig. 3.2.

3.3 Learning Local Taxonomic Relation Classifier

The TAREC training algorithm focuses on learning to predict the probability distribution over the possible taxonomic relations between two terms. It is clear that two single terms do not provide informative features to predict a relation between them. Our key idea is that we first map input terms to a more expressive representation space that allows us to extract rich features. To accommodate this idea, we take advantage of the structure of Wikipedia pages to map input terms to corresponding pages in Wikipedia.

Conceptually, Wikipedia provides a category structure. Thus, it may help us directly read off the taxonomic relations between terms. However, since terms could be ambiguous, this could lead to uncertain situations when they are mapped to Wikipedia pages (e.g. the term *Ford* could be mapped to both *Ford Motor Company* and president *Gerald Ford*.) Furthermore, even if a term is mapped to a Wikipedia page correctly, it is not easy to directly use the Wikipedia category system to infer its relation to another term due to the fact that the taxonomic relation information may be hidden in the text of their Wikipedia pages, not simply in the categories. For example, *Bill Clinton* is a descendant of *American*, but there is no explicit Wikipedia category *American* in the Wikipedia page of *Bill Clinton*. Nevertheless, the fact that there are indeed categories *American health activists* and *American humanitarians* on the *Bill Clinton* Wikipedia page would be very helpful to inferring its taxonomic relation to the term *American*.

In this section, we first briefly describe the structure of Wikipedia pages, then we introduce two mapping procedures that produce different behaviors in our final systems, and finally, we present

Page Title	Text	Categories
Regular (Non-Redirection) pages		
President of the United States	The President of the United States is the head of state and head of government of the United States and is the highest political official in the United States by influence and recognition. The President leads the executive branch of the federal government and is one of only two elected members of the executive branch...	Presidents of the United States, Presidency of the United States
George W. Bush	George Walker Bush; born July 6, 1946) served as the 43rd President of the United States from 2001 to 2009. He was the 46th Governor of Texas from 1995 to 2000 before being sworn in as President on January 20, 2001...	Children of Presidents of the United States, Governors of Texas, Presidents of the United States, Texas Republicans...
Gerald Ford	Gerald Rudolff Ford (born Leslie Lynch King, Jr.) (July 14, 1913 December 26, 2006) was the 38th President of the United States, serving from 1974 to 1977, and the 40th Vice President of the United States serving from 1973 to 1974.	Presidents of the United States, Vice Presidents of the United States, Republican Party (United States) presidential nominees...
Redirect pages		
US President	#Redirect [[President of the United States]]	(N/A)
Gerald R. Ford	#Redirect [[Gerald Ford]]	(N/A)
Disambiguation pages		
Ford	#Refer [[Ford Motor Company]] #Refer [[Gerald R. Ford]] #Refer [[Henry Ford]] #Refer [[Ford's Theatre]]	Disambiguation page, Surnames
table	#Refer [[Table (furniture)]] #Refer [[Table (information)]] #Refer [[Table (database)]]	Disambiguation page

Table 3.2: An excerpt of the structure of Wikipedia pages

the learning features.

3.3.1 The Structure of Wikipedia Pages

The majority of Wikipedia pages provide information about concepts (or entities). Typically, each concept page consists of three important pieces of information: a title (usually identical to the concept surface form), a body text which describes the concept, and the categories to which the

concept belongs. The upper part of Table 3.2 shows snippets of some regular pages exemplifying the information of concepts *President of the United States*, *George W. Bush* and *Gerald Ford*.

In addition, it is common for a concept to be referred to in multiple ways. For example, *Gerald Ford* can also be referred to as *Gerald R. Ford*, *Gerald Rudolph Ford, Jr.* or *President Ford*. Fixed resources, such as WordNet and the Extended WordNet, are not able to deal with this issue, whereas the Wikipedia page structure provides an excellent resource to address this problem. The reason is that Wikipedia maintains a huge system of redirect pages that redirects uncanonical concepts to their canonical form. The middle part of Table 3.2 illustrates some redirect pages and their references. Redirect pages usually do not have categories because the categories are maintained on corresponding canonical pages.

Furthermore, a term may be ambiguous and could refer to multiple concepts. Fortunately, Wikipedia provides a clear organization of ambiguous concepts in a special page structure that consists of disambiguation pages. Each disambiguation page contains several concepts that an ambiguous term may refer to. The last part of Table 3.2 shows the disambiguation pages of two terms, *Ford* and *table*, and the concepts they refer to. Note that it is possible for a referred concept to be linked to a redirect page. For instance, *Ford* may refer to *Gerald R. Ford*, which is, in turn, redirected to canonical the concept *Gerald Ford*.

Together, all these pieces of information make Wikipedia page structure a valuable resource when building a semantic representation for input terms.

3.3.2 Wikipedia-based Semantic Representation

In this section, we present two approaches to constructing Wikipedia-based semantic representations for input terms. Both approaches are motivated by the intuition that real-world applications are usually interested in identifying relation between related terms rather than arbitrary ones. For example, it is more likely that the term *Ford* in the pair (*George W. Bush*, *Ford*) refers to the former president of the United States, *Gerald Ford*, than to the car manufacturer *Ford Motor Company* or its founder *Henry Ford*. Our approaches below take this intuition into account when constructing the term's Wikipedia semantic representation.

In the following section, we use *Wikipedia concept* and *Wikipedia page* interchangeably to refer

to the Wikipedia page associated with a concept expressed by the page title. For example, the Wikipedia page *Gerald Ford* is associated with the Wikipedia concept *Gerald Ford*.

Match-based Approach

Intuitively, given a term, this approach looks for the most appropriate Wikipedia page that best matches (i.e. best describes) the term. To this end, the match-based approach maps the term to Wikipedia pages by directly looking it up and matching it with Wikipedia pages' title. Beside regular pages, we make use of both redirect and disambiguation pages. Given a pair of terms, the output of this procedure includes two Wikipedia pages that provide the best description of the two input terms, respectively.

In this approach, each Wikipedia concept page, p_x (where x is a Wikipedia concept), is represented by a set of keywords, KW_x. The keywords are extracted by selecting the top tokens in the body text and the categories of the page ranked by their TF-IDF scores. In this work, for each Wikipedia page, we use the first paragraph of the body text as an approximation for the whole text. We use the Porter stemmer to normalize the tokens.[3] For example, Wikipedia page *Gerald Ford* is represented by the following list of normalized tokens {*ford, presid, amend, gerald, vice, fifth, serv, fortieth, state, rudolph, nixon, unit, resign, eighth, thirti, constitut, twenti, term, person, bachil, episcopalian, adopte, waterg, wolverin, cardiovascular, nomine, recipi, communist, lawyer, rapid, omaha, scout, death, descend, yale, alumni, eagl*}. In our experiments, we used a maximum of top 40 keywords of KW_x, including the top 20 keywords of the text and the top 20 keywords from the categories of x.

Furthermore, each Wikipedia concept x is characterized by an absolute prominence score, α_x, which is defined as the number of times it is hyperlinked in the whole Wikipedia corpus. Intuitively, the prominence notion of a concept encodes its popularity by measuring how often it is linked to from other Wikipedia pages. Given a pair of unambiguous concepts (x, y), we define their similarity as follows:

$$sim(x, y) = \alpha_x \times \alpha_y \times |KW_x \cap KW_y|$$

[3] http://tartarus.org/~martin/PorterStemmer/

Match-based Algorithm

INPUT | A pair of terms (x, y).

ALGORITHM:

1. $Pool_x = \emptyset$; $Pool_y = \emptyset$
2. if x in $\mathcal{W}_{DP}$ // x is an ambiguous concept
3. $\quad DP_x = \mathcal{W}_{DP}(x)$
4. $\quad Pool_x \leftarrow \{$the concepts in $DP_x\}$
5. else if x in $\mathcal{W}_R$ // x is an unambiguous concept, but redirected
6. $\quad R_x = \mathcal{W}_R(x)$
7. $\quad Pool_x \leftarrow \{$the redirected concept in $R_x\}$
8. else if x in $\mathcal{W}_{NR}$ // x is an unambiguous (non-redirection) concept
9. $\quad NR_x = \mathcal{W}_{NR}(x)$
10. $\quad Pool_x \leftarrow \{NR_x\}$
11. Similarly, extract $Pool_y$ for y as from step 2 to 10.
12. Find the best pair of pages $(u^*, v^*) = argmax_{u \in Pool_x, v \in Pool_y} sim(u, v)$
13. RETURN: $\mathfrak{R}_x = \{u^*\}$ and $\mathfrak{R}_y = \{v^*\}$.

Figure 3.3: The Match-based approach to constructing Wikipedia-based semantic representations.

For a disambiguation page DP_x of term x (e.g. $x =$ *Ford* as shown in Table 3.2), each referred concept $u \in DP_x$ is assigned a relative prominence score $\alpha_u^x = \frac{\alpha_u}{max_{u' \in DP_x} \alpha_{u'}}$, where α_u is the absolute prominence scores of u. Given a concept $u \in DP_x$ and a concept $v \in DP_y$, we define the similarity score of pair (u, v) as follows: $sim(u, v) = \alpha_u^x \times \alpha_v^y \times |KW_u \cap KW_v|$. In general, if x is unambiguous (i.e. x matches a normal page or a redirected page in Wikipedia), its absolute prominence score is used. Otherwise, relative prominence score is used in the similarity metric.

Let $\mathcal{W}_{DP}$ be the list of Wikipedia disambiguation pages, $\mathcal{W}_R$ be the list of redirect pages, and $\mathcal{W}_{NR}$ be the list of regular (non-redirection) pages. We use $\mathcal{W}_{DP}(x)$, $\mathcal{W}_R(x)$ and $\mathcal{W}_{NR}(x)$ to denote the functions that map term x to the best corresponding Wikipedia page in $\mathcal{W}_{DP}$, $\mathcal{W}_R$ and $\mathcal{W}_{NR}$, respectively. A term is mapped to a Wikipedia page via an exact string matching operation between the term and the title of the page.

Given input pair (x, y), the match-based approach follows the algorithm sketched in Figure 3.3 to select the best Wikipedia page for each input term.

Note that if x is unambiguous, x is mapped to a single Wikipedia page, and $Pool_x$, therefore, has only one single member. In this case, the absolute prominence score α_x is used in the similarity scoring function. This is similar for y and $Pool_y$.

Search-based Approach

The key idea in our second approach is that we look for a set of relevant pages in the Wikipedia corpus to be used as a representation of a term, rather than a single page as in the match-based approach. This approach requires information retrieval techniques to search and retrieve relevant Wikipedia pages. In this work, we use the local search engine Lucene.[4] The main procedure of this approach proceeds as follows:

1. Input: A pair of well-segmented terms (x, y).
2. Create a unified query by concatenating x *AND* y. For example, for pair (*George W. Bush*, *Gerald Ford*), the unified query is *George W. Bush AND Gerald Ford*.
3. Search the complete Wikipedia corpus text using the unified query to retrieve a list of relevant pages, $\mathcal{L}_{x,y}$.
4. Extract the top important keywords from the categories of the pages in $\mathcal{L}_{x,y}$ by ranking them using TF-IDF scores. Intuitively, this search will retrieve relevant pages for both input terms, so the top extracted keywords will tie the semantic meaning of the two input terms to each other. For example, the unified query in step (1) will retrieve relevant pages of both *George W. Bush* and *Gerald Ford*. From the retrieved pages, extracted keywords may include: *president*, *politician*, *united*, *state*, etc..
5. Concatenate each input term with the list of keywords extracted in step 4. For instance, *George Bush* will be augmented to make a conjunctive query: *George W. Bush* AND *president* AND *politician* AND *united* AND *state*.
6. Search for the top relevant pages, $\mathfrak{R}_x$ and $\mathfrak{R}_y$ of x and y using their new queries from step 5.
7. Return: $\mathfrak{R}_x$ and $\mathfrak{R}_y$ as the Wikipedia representations of x and y, respectively.

In our experiments, we use the top 10 keywords in step 4, and 10 Wikipedia pages as the maximum number of pages in the Wikipedia representation of each term returned in step 7.

Term	Page Title	Text	Category
Gerald Ford	Gerald Ford	Gerald Rudolff Ford (born Leslie Lynch King, Jr.) (July 14, 1913 December 26, 2006) was the 38th President of the United States, serving from 1974 to 1977, and the 40th Vice President of the United States serving from 1973 to 1974...	Presidents of the United States, Vice Presidents of the United States, Republican Party (United States) presidential nominees...
	Presidency of Gerald Ford	Gerald Rudolff Ford (born Leslie Lynch King, Jr.) (July 14, 1913 December 26, 2006) was the 38th President of the United States, serving from 1974 to 1977, and the 40th Vice President of the United States serving from 1973 to 1974...	Presidents of the United States, Vice Presidents of the United States, Republican Party (United States) presidential nominees...
	Electoral history of Gerald Ford	Electoral history of Gerald Ford, 38th President of the United States and 40th Vice President of the United States...	Gerald Ford, Electoral history of American politicians...
Bush	George W. Bush	George Walker Bush; born July 6, 1946) served as the 43rd President of the United States from 2001 to 2009. He was the 46th Governor of Texas from 1995 to 2000 before being sworn in as President on January 20, 2001...	Children of Presidents of the United States, Governors of Texas, Presidents of the United States, Texas Republicans...
	George H. W. Bush	George Herbert Walker Bush (born June 12, 1924) is an American politician who served as the 41st President of the United States (198993)...	Parents of Presidents of the United States, Presidents of the United States, Texas Republicans...
	Presidency of George W. Bush	The presidency of George W. Bush began on January 20, 2001, when he was inaugurated as the 43rd President of the United States of America...	Presidencies of the United States, Presidency of George W. Bush...

Table 3.3: A short version of the Wikipedia representation of input pair *(Bush, Gerald Ford)*. Note that page *Presidency of Gerald Ford* is redirected to page *Gerald Ford*; they, therefore, get the same text and category list.

3.3.3 Feature Extraction

The features of a pair of terms are extracted from their Wikipedia representations. As discussed earlier (Section 3.3.1), a regular Wikipedia page of a Wikipedia concept usually consists of a title, a body text, and a list of categories to which the concept belongs. For convenience, for a term x, we use *the titles of* x, *the text of* x, and *the categories of* x to refer to the titles, text, and categories of the associated pages in the representation of x. Table 3.3 shows a short version of the Wikipedia representation of two input terms *Gerald Ford* and *Bush* extracted by the search-based approach. Note that the pages in the Wikipedia representation of *Bush* are mostly about presidency because the term is influenced by the other term *Gerald Ford*, as expected in the search-based approach. In this context, *the titles of Gerald Ford*, *the text of Gerald Ford* and *the categories of Gerald Ford* consist of all the titles, the body text and the categories of the Wikipedia pages in the Wikipedia representation of *Gerald Ford*, respectively. Similar notions apply to the term *Bush*.

In addition to the direct categories of a Wikipedia page of a term, we also collect its higher-level categories: we start from the categories of the page in its representation and recursively go up K levels on the Wikipedia category system as before. The categories of a term are the union of its direct categories and all the categories of the upper level pages.

Below we present the features extracted for an input pair of terms, (x, y), that will be used in learning the relations classifier. All features are real value.

Bag-of-words Similarity: We define four bag-of-words features as the degree of similarity among the texts and categories of x and y. The features are shown in Table 3.4. We use the cosine similarity metric to measure the value of these features. Let $v_x^t =< w_1, w_2, \cdots >$ be the bag-of-words feature vector of the texts of term x, where w_i is the indicator variable that indicates whether a particular word at position i is present in the text of x. Let $v_y^c =< w_1, w_2, \cdots >$ be the bag-of-words feature vector of the category of y. The similarity strength between the text of x and the categories of y is measured as in Equation (5.4).

$$sim(v_x^t, v_y^c) = \frac{\overrightarrow{v_x^t} \bullet \overrightarrow{v_y^c}}{\|\overrightarrow{v_x^t}\| \|\overrightarrow{v_y^c}\|} \tag{3.1}$$

[4]E.g. http://lucene.apache.org/

Bag-of-words similarity features
text(x) vs. categories(y)
categories(x) vs. text(y)
text(x) vs. text(y)
categories(x) vs. categories(y)

Table 3.4: Bag-of-word similarity features of (x,y), where texts(*term*) and categories(*term*) are the functions that extract associated texts and categories from the semantic representation of *term*.

For the other three bag-of-words features, we use similar notions and formulas.

Association Information: This feature measures the association information between terms by considering their information overlap over the whole Wikipedia data. We capture this feature using pointwise mutual information (*PMI*) which quantifies the discrepancy between the probability of two terms appearing together versus the probability of each term appearing independently.[5] The PMI of two terms x and y is estimated as in Equation (5.5):

$$PMI(x,y) = log\frac{p(x,y)}{p(x)p(y)} = log\frac{Nf(x,y)}{f(x)f(y)} \tag{3.2}$$

where N is the total number of Wikipedia pages, and f is a counting function that returns the number of times its argument(s) appear(s) (together) in Wikipedia.

Overlap Ratios: The overlap ratio features capture the fact that the titles of an ancestor term usually overlap with the categories of its descendants. Similarly, the categories of two sibling terms are also usually highly overlapping. For example, Wikipedia page *Presidents of the United States*, as shown in Table 3.2, has a title that overlaps with one of the categories of the Wikipedia page *George W. Bush*. This evidence strongly supports the conclusion that the term *Presidents of the United States* is an ancestor of the term *George W. Bush*. On the other hand, the categories of *George W. Bush* and *Gerald Ford* overlap with each other in several categories, such as *Presidents of the United States*. In general, a higher overlap ratio indicates a better chance for two terms to hold a taxonomic relation. We use three overlap ratio features as shown in Table 3.5.

We measure the overlap ratios by the ratios of the numbers of *key phrases* in the titles and

[5] *PMI* is different than mutual information. The former applies to specific outcomes, while the latter is used to measure the mutual dependence of two random variables.

Overlap ratio features
titles(x) vs. categories(y)
categories(x) vs. titles(y)
categories(x) vs. categories(y)

Table 3.5: Overlap ratio features of (x,y), where titles(*term*) is a function that returns the titles of the Wikipedia pages in the Wikipedia representation of *term*; function categories(*term*) was defined in Table 3.4

categories of the input terms. In our context, a phrase is considered to be a key phrase if it belongs to one of the following types:

- the whole string of a title or category
- the lemma of the head a category
- the post-modifier of a category

We use the Noun Group Parser [Suchanek et al., 2007] to extract the head and post-modifier of a category. For example, the category *Cities in Illinois* of Wikipedia page *Chicago* could be parsed into a head in its root form, *City*, and a post-modifier, *Illinois*. Therefore, in the pair of terms *(City, Chicago)*, the term *City* overlaps with the head, *City*, of the category *Cities in Illinois* of the Wikipedia page *Chicago*. This is a strong feature indicating that *Chicago* is a descendant of *City*.

Let two input terms be x and y. Let $u_x^t = (t_x^1, t_x^2, \cdots)$ denote the set of titles of term x in its Wikipedia representation. Also, let $u_y^c = (c_y^1, c_y^2, \cdots)$ be the set of the key phrases of the categories of term y in its Wikipedia representation. The overlap ratio feature between the titles of term x and the categories of term y is computed using the Jaccard similarity coefficient metric as shown in Equation (5.6).

$$overlap(x, y) = \frac{|u_x^t \bigcap u_y^c|}{|u_x^t \bigcup u_y^c|} \tag{3.3}$$

For the other two overlap ratio features, we use similar notions and formulas. In addition, to measure the overlap ratio feature between the categories of the two input terms, the post-modifiers of the categories are not used because when the categories of the terms are compared together, the overlap of the post-modifiers of the categories is not useful (e.g. categories *Actors of America* and

Companies of America overlap in their post-modifiers *America*, but this overlap does not help to recognize taxonomic relations).

Overall, we use eight feature types for the local classifier including: bag-of-words features (4), association information (1), and overlap ratio features (3).

3.3.4 Non-Wikipedia Terms

Although most commonly used terms have corresponding Wikipedia pages, new entities and concepts always come up and there are still many terms that do not have Wikipedia pages. We call these terms *non-Wikipedia terms*. In order to handle these terms, we propose to use a normalization procedure to find approximate Wikipedia pages for non-Wikipedia terms. The basic idea of the normalization procedure is to find a replacement for a non-Wikipedia term that, ideally, keeps the underlying taxonomic relation unchanged, by using Web search. For example, given input pair (*Lojze Kovačič*, *Rudi Šeligo*), there is no English Wikipedia page for *Lojze Kovačič*, who is a writer, but if we can find another writer, such as *Marjan Rožanc*, and use it as a replacement of *Lojze Kovačič*, then we can continue classifying the taxonomic relation of pair (*Marjan Rožanc*, *Rudi Šeligo*).

Our Wikipedia normalization procedure follows [Sarmento et al., 2007]. We first compose a query concatenating the two input terms (e.g. *Lojze Kovačič AND Rudi Šeligo*) and use Web search[6] to retrieve list-structure snippets with the following pattern: "... $\langle del \rangle$ c_a $\langle del \rangle$ c_b $\langle del \rangle$ c_c $\langle del \rangle$ ..." (the two input terms must be among c_a, c_b, c_c, ...). In the pattern, *del* is a delimiter and could be commas, periods, or asterisks.[7] Using the snippets that contain the patterns of interest, we extract c_a, c_b, c_c etc. as replacement candidates. To reduce noise, we empirically constrain the list to contain at least 4 terms that are no longer than 20 characters each.[8] The candidates are ranked based on their occurrence frequency. The top candidate for which we can construct a Wikipedia representation, is used as a replacement.

[6] http://developer.yahoo.com/search/web/

[7] Periods and asterisks capture enumerations.

[8] We believe that a list with less than 4 terms may not be a good list. Furthermore, we require that a candidate term has no more than 20 characters to prevent noisy terms.

3.4 Global Inference with Relational Constraints

In this section, we present a novel inference model which relies on the structure of pair-wise mutual taxonomic relations among two input terms and some additional related terms to enforce final coherent prediction. The main idea of our inference model is that logical constraints on relations among terms may prevent predicting illegitimate structures. Our global objective, therefore, focuses on selecting the best taxonomic relation between two input terms that allows legitimate structures to be formed when additional terms are taken into account. For example, given two target terms *George W. Bush* and *president*, we add an additional related term, such as *Bill Clinton*; if we can identify, with some degree of confidence, that (i) *president* is an ancestor of *Bill Clinton*, and (ii) *Bill Clinton* is a sibling of *George W. Bush*, then due to the transitivity property of taxonomic relations, the term *George W. Bush* is likely to be a descendant of the term *president* since other relations will create illegitimate structures.

The two input terms along with some additional related terms and the taxonomic relations among them form a structure that we call a *term network* (or *network* for short). Fig. 3.4 shows some n-term networks consisting of two input terms (x, y), and additional terms v, w, z. Note that the arrows in the figures follow the notions in Table 3.1.

The aforementioned observations suggest that if we can get additional terms that are related to the two input terms, we can enforce coherent structures and eliminate illegitimate combinations of terms and relations via relational constraints. This would help the system improve the predictions of taxonomic relations of input pairs. In this work, we formalize our inference model using constraint-based formulations that were introduced to the NLP community in [Roth and Yih, 2004] and were shown to be very effective in exploiting declarative background knowledge [Denis and Baldridge, 2007, Punyakanok et al., 2008, Chang et al., 2008b].

3.4.1 Enforcing Relational Constraints through Global Inference

The main goal of our inference model is to eliminate illegitimate term networks and select the best taxonomic relation of two input terms, embedded in legitimate structures. Below, we formalize our inference model with the following notation:

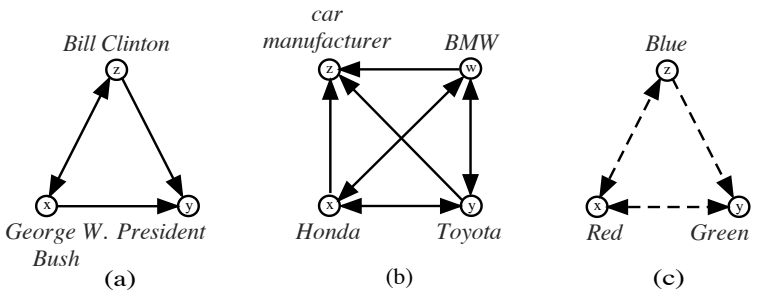

Figure 3.4: Examples of n-term networks with input pair (x, y). (a) and (b) show two valid structures, whereas (c) illustrates a relational constraints with an illegitimate structure.

- (x, y) : two input terms.
- $\mathcal{Z}_{x,y} = \{z_1, z_2, ..., z_m\}$: a set of additional terms.
- $Z \subseteq \mathcal{Z}_{x,y}$: a subset of terms in $\mathcal{Z}_{x,y}$.
- e : an edge imposing a relation between two terms; e can be one of four relations.
- $w(e)$: the weight of e, given by local classifier $\mathcal{C}$ (see Fig. 3.1). Recall that $\mathcal{C}$ predicts a probability distribution over four taxonomic relations between x and y.

Each network is formed by x, y and the terms in Z. Let $l = |Z|$, then there are $n = 2 + l$ terms in each network, and $4^{\left[\frac{1}{2}n(n-1)\right]}$ networks can be constructed.

We define a *relational constraint* as a network that imposes an *illegitimate structure* on its edges. That is, a constraint is *unlexicalized* in the sense that we only consider the edge structure of the network, regardless of the specific terms. In this work, we focus on 3-term networks (i.e. $l = 1$). For example, given input pair (*red*, *green*) and $\mathcal{Z} = \{blue, yellow\}$, we can construct 64 networks for triple $\langle red, green, Z = \{blue\}\rangle$ and 64 networks for $\langle red, green, Z = \{yellow\}\rangle$ by trying all possible relations between the terms.

Fig. 3.4(c) shows a relational constraint where the term *red* is a sibling of both *green* and *blue*, but *green* is an ancestor of *blue*; this structure is illegitimate because of the transitivity property. The relational constraints in this work are manually constructed. In the case of 3-term networks, constraints are written in a clockwise direction, starting from the two input terms, (x, y). For instance, the illegitimate structure in Fig. 3.4(c) forms the following relational constraint: $\langle \leftrightarrow, \leftrightarrow, \rightarrow \rangle$, where the arrows follow the notation in Table 3.1.

We solve this constraint optimization problem by a 2-stage greedy approach which is integrated into the Constrained Conditional Model (CCM) [Chang et al., 2008b]. We first check and eliminate all term networks that are illegitimate, then greedily select the best taxonomic relation that allows legitimate networks.[9]

Let $\mathcal{RC}$ be a list of relational constraints. Network t can be assigned a score using the network scoring function defined in Eq. (3.4). This scoring function is a linear combination of the edge weights, $w(e)$, of the edges in t and the penalties, ρ_k, that penalize if the edge structure of t belongs to $\mathcal{RC}$.

$$score(t) = \sum_{e \in t} w(e) - \sum_{k=1}^{|\mathcal{RC}|} \rho_k d_{\mathcal{RC}_k}(t) \tag{3.4}$$

where, function $d_{\mathcal{RC}_k}(t)$ indicates whether t matches $\mathcal{RC}_k$.

We can define relational constraints as either hard or soft constraints. In the current work, we consider illegitimate networks as hard constraints: all term networks that belong to the list of relational constraints are simply discarded. To do this, we set penalty factor ρ_k to ∞, for all $\mathcal{RC}_k$.

Now, among the set of networks formed by $\langle x, y, Z \rangle$, we select the best network as follows:

$$t^*_Z = \text{argmax}_t score(t) \tag{3.5}$$

Let $t^*_\cap = \cap_Z t^*_Z$, $Z \subseteq \mathcal{Z}$; we then partition $t^*_\cap$ into four groups according to the relation, denoted by rel, between x and y in each network. Let us denote each group by $\mathcal{T}_{rel}$. To choose the best taxonomic relation, rel^*, between x and y, we pick the relation which maximizes the average score of the whole group as in Eq. (3.6).

$$rel^* = \text{argmax}_{rel} \frac{1}{|\mathcal{T}_{rel}|} \sum_{t^* \in \mathcal{T}_{rel}} \lambda_{t^*} score(t^*) \tag{3.6}$$

where λ_t is the weight of the unlexicalized term network t, defined as the occurrence probability of t in an augmented version of the training data. To augment the training data, we first extract additional terms for each pair of terms in the training data, and then apply our local classifier to identify the taxonomic relation between the terms. The weight of a network t is computed as

[9] We do not use an exact inference approach (e.g. Integer Linear Programming (ILP)) to solve the problem because the optimization problem here with 3-term networks is small and can be effectively solved by a greedy approach. However, ILP and other optimization approaches could be used as the alternatives to our greedy approach.

Yago Query Patterns

INPUT: A term x

OUTPUT: Lists of ancestors, siblings, and children of x

Pattern 1	Pattern 2	Pattern 3
x MEANS ?A	x MEANS ?A	x MEANS ?D
?A SUBCLASSOF ?B	?A TYPE ?B	?E TYPE ?D
?C SUBCLASSOF ?B	?C TYPE ?B	

RETURN: ?B, ?C, ?E as lists of ancestors, siblings (extracted by Patterns 1 and 2), and children (extracted by Pattern 3), respectively.

Figure 3.5: Our YAGO query patterns used to obtain related terms for x.

the number of time t occurs in the augmented training data divides by the total number of term networks.

3.4.2 Extracting Related Terms

In the inference model, we need to obtain additional terms, $\mathcal{Z}_{x,y}$, that are related to x and y. Hereafter, we refer to additional terms as *related terms*. The related term space is composed of the direct ancestors, siblings and direct children of the input terms, obtained from some knowledge source.

We propose to extract related terms from the YAGO ontology [Suchanek et al., 2007]. YAGO is chosen over the Wikipedia category system used in our work because YAGO is a clean ontology built by carefully combining Wikipedia and WordNet.[10]

In the YAGO model, all objects (e.g. *cities*, *people*, etc.) are represented as *entities*. To map our input terms to entities in YAGO, we use the MEANS relation defined in the YAGO ontology. Furthermore, similar entities are grouped into *classes*. This allows us to obtain direct ancestors of an entity by using the TYPE relation which gives the entity's classes. Furthermore, we can get ancestors of a class with the SUBCLASSOF relation.[11] By using three relations, MEANS, TYPE and SUBCLASSOF, in the YAGO model, we can obtain direct ancestors, siblings, and direct children, if any, for input terms. In the case that the two input terms are not contained in YAGO, the inference model is simply ignored. Fig. 3.5 presents three patterns that we use to query related terms from

[10]However, YAGO by itself is weaker than our system in identifying taxonomic relations (see Section 3.5).

[11]These relations are defined in the YAGO ontology.

YAGO.

3.5 Experimental Study

In this section, we evaluate TAREC against other systems built upon existing well-known knowledge sources. The resources are either hierarchical structures or extracted by using distributional semantic models. We also provide experimental analyses on the compared systems.

3.5.1 Comparison to Hierarchical Structures

Data Preparation

We create and use two main data sets in these experiments.

Dataset-I is generated from 40 semantic classes (see Appendix A) of about 11,000 instances. The original semantic classes and instances were manually constructed with a limited amount of manual post-filtering and were used to evaluate information extraction tasks in [Paşca, 2007, Paşca and Van Durme, 2008] (we denote this original data as **OrgData-I**). This data set contains both terms with Wikipedia pages (e.g. *George W. Bush*) and non-Wikipedia terms (e.g. *hindu mysticism*). Pairs of terms are generated by randomly pairing semantic class names and instances. We generate disjoint training and test sets of 8,000 and 12,000 pairs of terms, respectively. We call the test set of this data set **Test-I**.

Dataset-II is generated from 44 semantic classes (see Appendix A) of more than 10,000 instances used in [Vyas and Pantel, 2009].[12] The original semantic classes and instances were extracted from Wikipedia lists. This data therefore contains only terms that have Wikipedia pages. We also generate disjoint training and test sets of 8,000 and 12,000 pairs of terms, respectively, and call the test set of this data set **Test-II**.

Both data sets contain both types of closed semantic classes (e.g. *chemical element*, *country*) and open semantic classes (e.g. *basic food*, *hurricane*). Moreover, there are classes with proper nouns (e.g. *actor* with *Mel Gibson*) and classes with common nouns (e.g. *basic food* with *rice*, *milk*).

[12]There were 50 semantic classes in the original data set. We grouped some semantically similar classes for the purpose of classifying taxonomic relations.

Many semantic class names in the original data sets are written in short forms. We expand these names to meaningful names that are used by all systems in our experiments. For example, *terroristgroup* is expanded to *terrorist group*, *terrorism*, *chemicalelem* to *chemical element*, *proglanguage* to *programing language*. Some examples are shown in Table 3.1. Four types of taxonomic relations are covered with balanced number of examples in all data sets.[13]

To evaluate our systems, we used a snapshot of Wikipedia from July, 2008. After cleaning and removing articles without categories (except redirect pages), 5,503,763 articles remained. We indexed these articles by their body texts using Lucene.[14] In practice, we only indexed the abstract (usually the first paragraph) of the Wikipedia pages. All characters were lower-cased and all punctuations were removed. We also removed stop words.[15] Furthermore, when performing search on the Wikipedia index, we did not normalize the search similarity score to the length of an article. Specifically, we overwrote the *lengthNorm* function in Lucene to always return value 1. All query tokens must occur in a Wikipedia page for it to be returned in the search result list.

We used the Regularized Averaged Perceptron [Freund and Schapire, 1999] as a learning algorithm within the LBJ modeling language [Rizzolo and Roth, 2010].[16] The learning algorithm used the one-vs-all scheme to transform a set of binary classifiers into a multi-class classifier. The raw activation scores were converted into probability distribution with the *softmax* function [Bishop, 1996]. If there are n classes and the raw score of class i is act_i, the posterior estimation for class i is:

$$Prob(i) = \frac{e^{act_i}}{\sum_{1 \le j \le n} e^{act_i}}$$

Compared Systems

Beside TAREC, we developed three other systems built upon well-known large-scale hierarchical structures.

Strube07 is built on the latest version of a taxonomy, T_{Strube}, which was derived from Wikipedia [Ponzetto and Strube, 2007]. It is worth noting that the structure of T_{Strube} is similar to the page

[13] Published at http://cogcomp.cs.illinois.edu/page/resources/TaxonomicRelationData.

[14] http://lucene.apache.org, version 2.3.2

[15] We used the following stop word list: *a, about, an, are, as, at, be, by, com, de, en, for, from, how, i, in, is, it, la, of, on, or, that, the, this, to, was, what, when, where, who, will, with, und, the, www.*

[16] http://cogcomp.cs.illinois.edu/page/software_view/11

structure of Wikipedia. For a fair comparison, we first generate a Wikipedia representation for each input term by following search-based approach in Section 3.3.2. The titles and categories of the articles in the representation of each input term are then extracted. Only titles and their corresponding categories that are in T_{Strube} are considered. A term is an ancestor of another one if at least one of its titles is in the categories of the other term. If two terms share a common category, they are considered siblings, otherwise they are considered to have no relation. The ancestor relation is checked first, then the sibling, and finally no relation.

Snow06 uses the Extended WordNet [Snow et al., 2006]. Words in the Extended WordNet can be common nouns or proper nouns. Given two input terms, we first map them onto the hierarchical structure of the extended WordNet by exact string matching. A term is an ancestor of another one if it can be found as a subsumer after recursively going up K levels in the hierarchical tree of the Extended WordNet from the other term. If two terms share a common subsumer within K levels on the tree, they are classified as siblings. Otherwise, there is no relation between them. Similar to Strube07, we first check ancestor, then sibling, and finally no relation.

Yago07 uses the YAGO ontology [Suchanek et al., 2007] as its main source of background knowledge. Because the YAGO ontology is a combination of Wikipedia and WordNet, this system is expected to perform well in identifying taxonomic relations. To access term's ancestors and siblings, we use patterns 1 and 2 in Fig. 3.5 to map a term to the ontology and move up on the ontology. The relation identification process is then similar to those of Snow06 and Strube07. If an input term is not recognized by these systems, they are considered to have no relation.

Our TAREC evaluation algorithm is described in Fig. 3.2 and is evaluated in two settings: **TAREC**^{MATCH}, which employs the match-based approach (Section 3.3.2), and **TAREC**^{SEARCH}, which uses the search-based approach (Section 3.3.2).

We evaluate each setting with the **Local** model which does classification on term pairs by directly selecting the highest-probability relation returned by the local classifier $\mathcal{C}$. For the **Inference** model, we manually construct a pre-defined list of 35 relational constraints (see Appendix B).

System	Test-I	Test-II
Strube07	24.32	25.63
Snow06	41.97	36.26
Yago07	65.93	70.63
Local		
TAREC^{MATCH}	79.64	77.56
TAREC^{SEARCH}	81.89	84.7
Inference		
TAREC^{SEARCH}	**85.34**	**86.98**

Table 3.6: Performance, in accuracy, of the systems on **Test-I** and **Test-II**. TAREC systems with local models simply use the local classifier to classify taxonomic relations by choosing the relation having highest confidence.

Results

In all systems compared, we vary the value of K, the deep of the Wikipedia category system that is examined in our approaches, from 1 to 4. The best results of the systems are reported.

Table 3.6 shows the comparison of all systems evaluated on both Test-I and Test-II. Our TAREC (Local) systems, as shown, significantly outperform the other systems. The results show that our machine learning-based classifier is very flexible in extracting features of the two input terms and is thus much better at predicting their taxonomic relation. In contrast, because other systems rely heavily on string matching techniques to map input terms to their respective ontologies, they are very inflexible and brittle. This clearly shows the limitations of using structured resources to classify taxonomic relations.

Between the local systems, the search-based approach is better than the match-based approach. This can be explained by the fact that the match-based approach is still not flexible enough in mapping input terms to Wikipedia representation.

We apply the inference model on top of TAREC^{SEARCH} (Local) and further achieve remarkable improvement. The improvement of TAREC^{SEARCH} (Inference) over TAREC^{SEARCH} (local) on Test-I shows the contribution of both the normalization procedure (see Section 3.3.4) and the global inference model to the classification decisions, whereas the improvement on Test-II emphasizes only the contribution of the inference model, because Test-II only contains terms that have corresponding Wikipedia pages. This improvement also suggests that relational constraints help improve the local classifier by enforcing coherent decisions over underlying structures of terms and relations.

Furthermore, it is also interesting to see that between Test-I and Test-II test sets, TAREC^{MATCH} (Local) performs better on Test-I. Our analysis shows that this is because there are more ambiguous terms (i.e. requiring more mappings to concepts in disambiguation pages) in Test-II than Test-I, therefore, Test-II is more difficult than Test-I for the match-based approach. Specifically, 36.23% of terms in Test-II are ambiguous, while that number in Test-I is 31.71%.

For the value of K, the best results of the systems on Test-I are achieved with: $K = 4$ for Strube07, $K = 2$ for Snow06, $K = 1$ for Yago07, $K = 3$ for TAREC^{MATCH}, and $K = 2$ for both local and inference models of TAREC^{SEARCH}. These values of K are the same on Test-II.[17] This shows that while the best value of K may vary with different systems, it is consistent across the data sets. Hence, we use $K = 2$ for further experiments with TAREC^{SEARCH}, unless specified otherwise.

We do not use special tactics to handle polysemous terms. However, our approaches to building Wikipedia representations for input terms described in Section 3.3 tie the senses of the two input terms together, thus, implicitly, tend to capture the potential meanings of the terms. We do not use this procedure in Snow06 because WordNet and Wikipedia are different in their structures. We also do not use this procedure in Yago07 because in YAGO, a term is mapped onto the ontology by using the MEANS operator (in Pattern 1, Fig. 3.5). This cannot follow our procedure.

3.5.2 Comparison to Harvested Knowledge

As we have discussed earlier, the outputs of bootstrapping-based algorithms is usually limited to a small number of high-quality terms while sacrificing coverage (or vice versa). For example, the full Espresso algorithm [Pantel and Pennacchiotti, 2006] extracted 69,156 instances of *is-a* relation with 36.2% of precision. Similarly, (Kozareva et al., 2008) evaluated only a small number (a few hundreds) of harvested instances. Recently, [Baroni and Lenci, 2010] proposed a general framework for extracting properties of input terms. Their **TypeDM** model harvested 5,000 significant properties for each term out of 20,410 noun mentions. For example, the properties of *marine* include $\langle own, bomb\rangle$, $\langle use, gun\rangle$. Using vector space models we could measure the similarity between terms using their property vectors. However, since the information available in TypeDM does not directly

[17] The best results on Test-II with $K = 2$ and $K = 3$ are similar.

support predicting ancestor relation between terms, we only evaluate TypeDM in classifying sibling vs. no relation. To accommodate this experiment, we develop the following procedure, giving a list of semantic classes.

- For each semantic class, use some seeds to compute a centroid vector from the seeds' vectors in TypeDM.
- Each term in an input pair is classified into its best semantic class based on the cosine similarity between its vector and the centroid vector of the semantic classes.
- Two terms are siblings if they are classified to the same semantic class; no relation, otherwise.

Out of the terms in OrgData-I, only 345 terms are covered by the noun mentions in TypeDM. These terms belong to 10 significant semantic classes. For each semantic class, we randomly pick 5 instances as its seeds to compute its single centroid vector. The rest of the overlapping instances are randomly paired to make a data set of 4,000 pairs of terms balanced in the number of sibling and no relation pairs. On this data set, TypeDM achieves an accuracy of 79.75%. TAREC^{SEARCH} (Local), with the local classifier trained on the training set (with 4 taxonomic relation classes) of Dataset-I, gives 78.35% of accuracy. TAREC^{SEARCH} (Inference) system achieves 82.65%. We also re-train and evaluate the local classifier of TAREC^{SEARCH} (Local) on the same training set but without ancestor-relation examples. This local classifier achieves an accuracy of 81.08%.

These results show that although the full system, TAREC^{SEARCH} (Inference), achieves better performance, TypeDM is very competitive in recognizing sibling vs. no relation. It has not been straightforward to apply the TypeDM model to ancestor relations between terms. As a result we only tested it in the limited setting where semantic classes are given in advance.

3.5.3 Experimental Analysis

In this section, we discuss some experimental analyses to better understand our systems. In all these experiments, TAREC uses the search-based approach to build Wikipedia representation.

Precision and Recall: We study TAREC on individual taxonomic relations using Precision and Recall. Table 3.7 shows that TAREC (Inference) performs very well on ancestor relations. Sibling and no relation are the most difficult relations to classify. In the same experimental setting

	Test-I		Test-II	
	Prec	Rec	Prec	Rec
$x \leftarrow y$	95.82	88.01	96.46	88.48
$x \rightarrow y$	94.61	89.29	96.15	88.86
$x \leftrightarrow y$	79.23	84.01	83.15	81.87
$x \nleftrightarrow y$	73.94	79.9	75.54	88.27
Average	85.9	85.3	87.83	86.87

Table 3.7: Performance of TAREC (Inference) on individual taxonomic relation.

System	Wiki	WordNet	non-Wikipedia
Strube07	24.59	24.13	21.18
Snow06	41.23	46.91	34.46
Yago07	69.95	70.42	34.26
TAREC (Local)	89.37	89.72	31.22
TAREC (Inference)	**91.03**	**91.2**	**45.21**

Table 3.8: Performance of the systems on special data sets, in accuracy. On the non-Wikipedia test set, TAREC (Local) simply returns sibling relation. Note that TAREC uses search-based approach to build Wikipedia representation for input terms.

on Test-I, Yago07 achieves 79.34% and 66.03% of average Precision and Recall, respectively. These numbers on Test-II are 81.33% and 70.44%.

Special Data Sets: We evaluate all systems that use hierarchical structures as background knowledge on three special data sets derived from Test-I. From 12,000 pairs in Test-I, we created a test set, **Wiki**, consisting of $10,456$ pairs with all terms in Wikipedia. We use the rest of $1,544$ pairs with at least one non-Wikipedia term to build a **non-Wiki** test set. The third data set, **WordNet**, contains $8,625$ pairs with all terms in WordNet and Wikipedia. Table 3.8 shows the performance of the systems on these data sets. Unsurprisingly, Yago07 gets better results on Wiki than on Test-I. Snow06, as expected, gives better performance on the WordNet test set. TAREC (Inference) still significantly outperforms these systems. The improvement of TAREC (Inference) over TAREC (local) on the Wiki and WordNet test sets emphasizes the contribution of the inference model, whereas the improvement on the non-Wikipedia test set shows the contribution of the normalization procedure described in Section 3.3.4.

Contribution of Related Terms in Inference: We evaluate TAREC (Inference) when the inference procedure is fed by related terms that are generated using a "gold standard" source instead of YAGO. To do this, we use the original data which was used to generate Test-I. For

System	K=1	K=2	K=3	K=4
TAREC (Inference)	82.93	85.34	85.23	83.95
TAREC (Gold Inference)	83.46	86.18	85.9	84.93

Table 3.9: TAREC with different sources providing related terms for inference.

each term in the examples of Test-I, we get its ancestors, siblings, and children, if any, from the original data and use them as related terms in the inference model. This system is referred to as **TAREC (Gold Inference)**. Table 3.9 shows the results of the two systems on different K as the number of levels to go up on the Wikipedia category system. We see that TAREC gets better results when doing inference with better related terms. In this experiment, the two systems use the same number of related terms.

3.6 Related Work

There are several works that aim at building taxonomies and ontologies which organize concepts and their taxonomic relations into hierarchical structures. [Snow et al., 2005, Snow et al., 2006] constructed classifiers to identify hypernym relationship between mentions from dependency trees of large corpora. Mentions with recognized hypernym relation are extracted and incorporated into a manually constructed lexical database, WordNet [Fellbaum, 1998], resulting in the Extended WordNet, which has been augmented this way with more than $400,000$ synsets.

In the work of [Ponzetto and Strube, 2007] and [Suchanek et al., 2007], the authors mined Wikipedia to construct hierarchical structures of concepts and relations. While the former exploited the Wikipedia category system as a conceptual network and extracted a taxonomy consisting of subsumption relations, the latter presented the YAGO ontology, which was automatically constructed by mining and combining Wikipedia structure and information with WordNet. A natural way to use these hierarchical structures to support taxonomic relation classification is to map targeted terms onto the hierarchies and check if they subsume each other or share a common subsumer. However, this approach is limited because constructed hierarchies may suffer from noise and inflexibility in dealing with ambiguous terms.

On the other hand, information extraction bootstrapping algorithms, such as described in [Pantel and Pennacchiotti, 2006, Kozareva et al., 2008], automatically harvest related terms on

large corpora by starting with a few seeds of pre-specified relations (e.g. *is-a*, *part-of*). Bootstrapping algorithms rely on some scoring function to assess the quality of terms and additional patterns extracted during bootstrapping iterations. Similarly, but with a different focus, Open IE, [Banko and Etzioni, 2008, Davidov and Rappoport, 2008b], deals with a large number of relations which are not pre-specified. Either way, the output of these algorithms is usually limited to a small number of high-quality terms while sacrificing coverage (or vice versa). Moreover, an Open IE system cannot control the extracted relations and this is essential when identifying taxonomic relations. Recently, there has been much work on distributional semantic models (DSMs) that leverage the context a word appears in to harvest words based on their semantic similarity in vector spaces [Padó and Lapata, 2007, Turney and Pantel, 2010, Baroni and Lenci, 2010]. Especially, [Baroni and Lenci, 2010] described a general framework of DSMs that extracts significant contexts of given terms from large corpora. Consequently, a term can be represented by a vector of contexts in which it frequently appears. Any vector space model could then use the terms' vectors to cluster terms into semantic classes. Sibling terms (e.g. *Honda*, *Toyota*), therefore, have very high chance to be clustered together. Nevertheless, this approach cannot recognize ancestor relations. In this work, we compare TAREC with this framework only on recognizing sibling vs. no relation, in a strict experimental setting which pre-specifies the semantic classes to which the terms belong.

3.7 Summary

We studied an important component of many computational linguistics tasks: determining taxonomic relations between terms. We have argued that simply looking up the relation of input terms in structured resources cannot support this task well enough, and provided empirical support for this claim. We presented TAREC, a novel algorithmic approach that leverages information from the Wikipedia structure and uses machine learning and a constraint-based inference model to mitigate the noise and the level of uncertainty inherent in these resources. Our experimental study showed that both the local and the global models of TAREC significantly outperform other systems built upon existing well-known knowledge sources. Moreover, our algorithmic approach generalizes and handles well non-Wikipedia terms across semantic classes. Our future work will include an evaluation of TAREC in the context of textual inference applications.

Chapter 4

Event Relation Discovery

This work develops a minimally supervised approach, based on focused distributional similarity methods and discourse connectives, for identifying of causality relations between events in context. While it has been shown that distributional similarity can help identifying causality, we observe that discourse connectives and the particular discourse relation they evoke in context provide additional information towards determining causality between events. We show that combining discourse relation predictions and distributional similarity methods in a global inference procedure provides additional improvements towards determining event causality.

4.1 Introduction

An important part of text understanding arises from understanding the semantics of events described in the narrative, such as identifying the events that are mentioned and how they are related semantically. For instance, when given a sentence "The police arrested him because he killed someone.", humans understand that there are two events, triggered by the words "arrested" and "killed", and that there is a causality relationship between these two events. Besides being an important component of discourse understanding, automatically identifying causal relations between events is important for various natural language processing (NLP) applications such as question answering, etc. In this work, we automatically detect and extract causal relations between events in text.

Despite its importance, prior work on event causality extraction in context in the NLP literature is relatively sparse. In [Girju, 2003], the author used noun-verb-noun lexico-syntactic patterns to learn that "mosquitoes cause malaria", where the *cause* and *effect* mentions are nominals and not necessarily event evoking words. In [Sun et al., 2007], the authors focused on detecting causality between search query pairs in temporal query logs. [Beamer and Girju, 2009] tried to detect causal

relations between verbs in a corpus of screen plays, but limited themselves to consecutive, or adjacent verb pairs. In [Riaz and Girju, 2010], the authors first cluster sentences into topic-specific scenarios, and then focus on building a dataset of causal text spans, where each span is headed by a verb. Thus, their focus was not on identifying causal relations between events in a given text document.

In this work, given a text document, we first identify events and their associated arguments. We then identify causality or relatedness relations between event pairs. To do this, we develop a minimally supervised approach using focused distributional similarity methods, such as co-occurrence counts of events collected automatically from an unannotated corpus, to measure and predict existence of causality relations between event pairs. Then, we build on the observation that discourse connectives and the particular discourse relation they evoke in context provide additional information towards determining causality between events. For instance, in the example sentence provided at the beginning of this section, the words "arrested" and "killed" probably have a relatively high apriori likelihood of being casually related. However, knowing that the connective "because" evokes a contingency discourse relation between the text spans "The police arrested him" and "he killed someone" provides further evidence towards predicting causality. The contributions of this work are summarized below:

- Our focus is on identifying causality between event pairs in context. Since events are often triggered by either verbs (e.g. "attack") or nouns (e.g. "explosion"), we allow for detection of causality between verb-verb, verb-noun, and noun-noun triggered event pairs. To the best of our knowledge, this formulation of the task is novel.

- We developed a minimally supervised approach for the task using focused distributional similarity methods that are automatically collected from an unannotated corpus. We show that our approach achieves better performance than two approaches: one based on a frequently used metric that measures association, and another based on the effect-control-dependency (ECD) metric described in a prior work [Riaz and Girju, 2010].

- We leverage on the interactions between event causality prediction and discourse relations prediction. We combine these knowledge sources through a global inference procedure, which

we formalize via an Integer Linear Programming (ILP) framework as a constraint optimization problem [Roth and Yih, 2004]. This allows us to easily define appropriate constraints to ensure that the causality and discourse predictions are coherent with each other, thereby improving the performance of causality identification.

4.2 Event Causality

In this work, we define an event as an action or occurrence that happens with associated participants or arguments. Formally, we define an event e as: $p(a_1, a_2, \ldots, a_n)$, where the predicate p is the word that triggers the presence of e in text, and $a_1, a_2, \ldots, a_n$ are the arguments associated with e. Examples of predicates could be *verbs* such as "attacked", "employs", *nouns* such as "explosion", "protest", etc., and examples of the arguments of "attacked" could be its *subject* and *object* nouns.

To measure the causality association between a pair of events e_i and e_j (in general, e_i and e_j could be extracted from the same or different documents), we should use information gathered about their predicates and arguments. A simple approach would be to directly calculate the pointwise mutual information (PMI)[1] between $p^i(a_1^i, a_2^i, \ldots, a_n^i)$ and $p^j(a_1^j, a_2^j, \ldots, a_m^j)$. However, this leads to very sparse counts as the predicate p^i with its list of arguments $a_1^i, \ldots, a_n^i$ would rarely co-occur (within some reasonable context distance) with predicate p^j and its entire list of arguments $a_1^j, \ldots, a_m^j$. Hence, in this work, we measure causality association using three separate components and focused distributional similarity methods collected about event pairs as described in the rest of this section.

4.2.1 Cause-Effect Association

We measure the causality or cause-effect association (CEA) between two events e_i and e_j using the following equation:

$$CEA(e_i, e_j) = s_{pp}(e_i, e_j) + s_{pa}(e_i, e_j) + s_{aa}(e_i, e_j) \tag{4.1}$$

where s_{pp} measures the association between event predicates, s_{pa} measures the association

[1] PMI is frequently used to measure association between variables.

between the predicate of an event and the arguments of the other event, and s_{aa} measures the association between event arguments. In our work, we regard each event e as being triggered and rooted at a predicate p.

Predicate-Predicate Association

We define s_{pp} as follows:

$$s_{pp}(e_i, e_j) = PMI(p^i, p^j) \times \max(u^i, u^j) \times IDF(p^i, p^j) \times Dist(p^i, p^j) \tag{4.2}$$

which takes into account the PMI between predicates p^i and p^j of events e_i and e_j respectively, as well as various other pieces of information. In Suppes' *Probabilistic theory of Casuality* [Suppes, 1970], he highlighted that event e is a possible cause of event e', if e' happens more frequently with e than by itself, i.e. $P(e'|e) > P(e')$. This can be easily rewritten as $\frac{P(e,e')}{P(e)P(e')} > 1$, similar to the definition of PMI:

$$PMI(e, e') = log\frac{P(e, e')}{P(e)P(e')}$$

which is only positive when $\frac{P(e,e')}{P(e)P(e')} > 1$.

Next, we build on the intuition that event predicates appearing in a large number of documents are probably not important or discriminative. Thus, we penalize these predicates when calculating s_{pp} by adopting the inverse document frequency (idf):

$$IDF(p^i, p^j) = idf(p^i) \times idf(p^j) \times idf(p^i, p^j),$$

where $idf(p) = log\frac{D}{1+N}$, D is the total number of documents in the collection and N is the number of documents that p occurs in.

We incorporate the distance measure of Leacock and Chodorow [Leacock and Chodorow, 1998], which was originally used to measure similarity between concepts, to also award event pairs that

are closer together, while penalizing event pairs that are further apart in texts:

$$Dist(p^i, p^j) = -log\frac{|sent(p^i) - sent(p^j)| + 1}{2 \times ws},$$

where $sent(p)$ gives the sentence number (index) in which p occurs and ws indicates the window-size (of sentences) used. If p^i and p^j are drawn from the same sentence, the numerator of the above fraction will return 1. In our work, we set ws to 3 and thus, if p^i occurs in sentence k, the furthest sentence that p^j will be drawn from, is sentence $k + 2$.

The final component of Equation 4.2, $\max(u^i, u^j)$, takes into account whether predicates (events) p^i and p^j appear most frequently with each other. u^i and u^j are defined as follows:

$$u^i = \frac{P(p^i, p^j)}{\max_k[P(p^i, p^k)] - P(p^i, p^j) + \epsilon}$$
$$u^j = \frac{P(p^i, p^j)}{\max_k[P(p^k, p^j)] - P(p^i, p^j) + \epsilon},$$

where we set $\epsilon = 0.01$ to avoid zeros in the denominators. u^i will be maximized if there is no other predicate p^k having a higher co-occurrence probability with p^i, i.e. $p^k = p^j$. u^j is treated similarly.

Predicate-Argument and Argument-Argument Association

We define s_{pa} as follows:

$$s_{pa}(e_i, e_j) = \frac{1}{|A_{e_j}|}\sum_{a \in A_{e_j}} PMI(p^i, a) + \frac{1}{|A_{e_i}|}\sum_{a \in A_{e_i}} PMI(p^j, a), \tag{4.3}$$

where A_{e_i} and A_{e_j} are the sets of arguments of e_i and e_j respectively.

Finally, we define s_{aa} as follows:

$$s_{aa}(e_i, e_j) = \frac{1}{|A_{e_i}||A_{e_j}|}\sum_{a \in A_{e_i}}\sum_{a' \in A_{e_j}} PMI(a, a') \tag{4.4}$$

Together, s_{pa} and s_{aa} provide additional contexts and robustness (in addition to s_{pp}) for measuring the cause-effect association between events e_i and e_j.

Our formulation of CEA is inspired by the ECD metric defined in [Riaz and Girju, 2010]:

$$ECD(a,b) = \max(v,w) \times -log\frac{dis(a,b)}{2 \times maxDistance}, \tag{4.5}$$

where

$$v = \frac{P(a,b)}{P(b) - P(a,b) + \epsilon} \times \frac{P(a,b)}{\max_t[P(a,b_t)] - P(a,b) + \epsilon}$$
$$w = \frac{P(a,b)}{P(a) - P(a,b) + \epsilon} \times \frac{P(a,b)}{\max_t[P(a_t,b)] - P(a,b) + \epsilon},$$

where ECD(a,b) measures the causality between two events a and b (headed by verbs), and the second component in the ECD equation is similar to $Dist(p^i, p^j)$. In our experiments, we will evaluate the performance of ECD against our proposed approach.

So far, our definitions in this section are generic and allow for any list of event argument types. In this work, we focus on two argument types: agent (subject) and patient (object), which are typical core arguments of any event. We describe how we extract event predicates and their associated arguments in the section below.

4.3 Verbal and Nominal Predicates

We consider that events are not only triggered by verbs but also by nouns. For a verb (verbal predicate), we extract its subject and object from its associated dependency parse. On the other hand, since events are also frequently triggered by nominal predicates, it is important to identify an appropriate list of event triggering nouns. In our work, we gathered such a list using the following approach:

- We first gather a list of deverbal nouns from the set of most frequently occurring (in the Gigaword corpus) 3,000 verbal predicate types. For each verb type v, we go through all its WordNet[2] senses and gather all its derivationally related nouns $\mathcal{N}_v$ [3].

- From $\mathcal{N}_v$, we heuristically remove nouns that are less than three characters in length. We

[2] http://wordnet.princeton.edu/

[3] The WordNet resource provides derivational information on words that are in different syntactic (i.e. part-of-speech) categories, but having the same root (lemma) form and that are semantically related.

also remove nouns whose first three characters are different from the first three characters of v. For each of the remaining nouns in $\mathcal{N}_v$, we measured its Levenstein (edit) distance from v and keep the noun(s) with the minimum distance. When multiple nouns have the same minimum distance from v, we keep all of them.

- To further prune the list of nouns, we next removed all nouns ending in "er", "or", or "ee", as these nouns typically refer to a person, e.g. "writer", "doctor", "employee". We also remove nouns that are not hyponyms (children) of the first WordNet sense of the noun "event"[4].

- Since we are concerned with nouns denoting events, FrameNet [Ruppenhofer et al., 2010] (FN) is a good resource for mining such nouns. FN consists of frames denoting situations and events. As part of the FN resource, each FN frame consists of a list of lexical units (mainly verbs and nouns) representing the semantics of the frame. Various frame-to-frame relations are also defined (in particular the *inheritance* relation). Hence, we gathered all the children frames of the FN frame "Event". From these children frames, we then gathered all their noun lexical units (words) and add them to our list of nouns. Finally, we also add a few nouns denoting natural disaster from Wikipedia[5].

Using the above approach, we gathered a list of about 2,000 noun types. This current approach is heuristics based which we intend to improve in the future, and any such improvements should subsequently improve the performance of our causality identification approach.

Event triggering deverbal nouns could have associated arguments (for instance, acting as subject, object of the deverbal noun). To extract these arguments, we followed the approach of [Gurevich et al., 2008]. Briefly, the approach uses linguistic patterns to extract subjects and objects for deverbal nouns, using information from dependency parses. For more details, we refer the reader to [Gurevich et al., 2008].

[4]The first WordNet sense of the noun "event" has the meaning: "something that happens at a given place and time"

[5]http://en.wikipedia.org/wiki/Natural_disaster

Coarse-grained	Fine-grained
Comparison	Concession, Contrast, Pragmatic-concession, Pragmatic-contrast
Contingency	Cause, Condition, Pragmatic-cause, Pragmatic-condition
Expansion	Alternative, Conjunction, Exception, Instantiation, List, Restatement
Temporal	Asynchronous, Synchronous

Table 4.1: Coarse-grained and fine-grained discourse relations.

4.4 Discourse and Causality

Discourse connectives are important for relating different text spans, helping us to understand a piece of text in relation to its context:

> [The police arrested him] because [he killed someone].

In the example sentence above, the discourse connective ("because") and the discourse relation it evokes (in this case, the *Cause* relation) allows readers to relate its two associated text spans, "The police arrested him" and "he killed someone". Also, notice that the verbs "arrested" and "killed", which *cross* the two text spans, are causally related. To aid in extracting causal relations, we leverage on the identification of discourse relations to provide additional contextual information.

To identify discourse relations, we use the Penn Discourse Treebank (PDTB) [Prasad et al., 2007], which contains annotations of discourse relations in context. The annotations are done over the Wall Street Journal corpus and the PDTB adopts a predicate-argument view of discourse relations. A discourse connective (e.g. because) takes two text spans as its arguments. In the rest of this section, we briefly describe the discourse relations in PDTB and highlight how we might leverage them to aid in determining event causality.

4.4.1 Discourse Relations

PDTB contains annotations for four coarse-grained discourse relation types, as shown in the left column of Table 4.1. Each of these are further refined into several fine-grained discourse relations, as shown in the right column of the table.[6] Next, we briefly describe these relations, highlighting those that could potentially help to determine event causality.

[6]PDTB further refines these fine-grained relations into a final third level of relations, but we do not use them in this work.

Comparison A *Comparison* discourse relation between two text spans highlights prominent differences between the situations described in the text spans. An example sentence is:

Contrast: [According to the survey, $x\%$ of Chinese Internet users prefer Google] whereas [$y\%$ prefer Baidu].

According to the PDTB annotation manual [Prasad et al., 2007], the truth of both spans is independent of the established discourse relation. This means that the text spans are not causally related and thus, the existence of a *Comparison* relation should imply that there is no causality relation across the two text spans.

Contingency A *Contingency* relation between two text spans indicates that the situation described in one text span causally influences the situation in the other. An example sentence is:

Cause: [The first priority is search and rescue] because [many people are trapped under the rubble].

Existence of a *Contingency* relation potentially implies that there exists at least one causal event pair crossing the two text spans. The PDTB annotation manual states that while the *Cause* and *Condition* discourse relations indicate casual influence in their text spans, there is no causal influence in the text spans of the *Pragmatic-cause* and *Pragmatic-condition* relations. For instance, *Pragmatic-condition* indicates that one span provides the context in which the description of the situation in the other span is relevant; for example:

Pragmatic-condition: If [you are thirsty], [there's beer in the fridge].

Hence, there is a need to also identify fine-grained discourse relations.

Expansion Connectives evoking *Expansion* discourse relations expand the discourse, such as by providing additional information, illustrating alternative situations, etc. An example sentence is:

Conjunction: [Over the past decade, x women were killed] and [y went missing].

Most of the *Expansion* fine-grained relations (except for *Conjunction*, which could connect arbitrary pieces of text spans) should not contain causality relations across its text spans.

Temporal These indicate that the situations described in the text spans are related temporally. An example sentence is:

Synchrony: [He was sitting at his home] when [the whole world started to shake].

Temporal precedence of the (cause) event over the (effect) event is a necessary, but not sufficient requisite for causality. Hence by itself, *Temporal* relations are probably not discriminative enough for determining event causality.

4.4.2 Discourse Relation Extraction System

Our work follows the approach and features described in the state-of-the-art Ruby-based discourse system of [Lin et al., 2010], to build an in-house Java-based discourse relation extraction system.

The system first identifies all discourse connective candidates in a given text. For each candidate, a discourse connective is used to recognize whether it is a a discourse connective or not. After that two text spans connected by each connective are detected. Finally, the core discourse classifier will classify the discourse relation evoked by the connective.

Our system only identifies explicit connectives in text. Similar to [Lin et al., 2010], we achieved a competitive performance of slightly over 80% F1-score in identifying fine-grained relations for explicit connectives. Our system is developed using the Learning Based Java modeling language (LBJ) [Rizzolo and Roth, 2010].

In the example sentences given thus far in this section, all the connectives were explicit, as they appear in the texts. PDTB also provides annotations for implicit connectives, which we do not use in this work. Identifying implicit connectives is a harder task and incorporating these is a possible future work.

4.5 Joint Inference for Causality Extraction

To exploit the interactions between event pair causality extraction and discourse relation identification, we define appropriate constraints between them, which can be enforced through the Constrained Conditional Models framework (aka ILP for NLP) [Roth and Yih, 2007, Chang et al., 2008a]. In doing this, the predictions of CEA (Section 4.2.1) and the discourse system are forced to cohere

with each other. More importantly, this should improve the performance of using only CEA to extract causal event pairs. To the best of our knowledge, this approach for causality extraction is novel.

4.5.1 CEA & Discourse: Implementation Details

Let $\mathcal{E}$ denote the set of event mentions in a document. Let $\mathcal{EP} = \{(e_i, e_j) \in \mathcal{E} \times \mathcal{E} \mid e_i \in \mathcal{E}, e_j \in \mathcal{E}, i < j, |sent(e_i) - sent(e_j)| \leq 2\}$ denote the set of event mention pairs in the document, where $sent(e)$ gives the sentence number in which event e occurs. Note that in this work, we only extract event pairs that are at most two sentences apart. Next, we define $\mathcal{L}_{ER} = \{$"causal", "¬ causal"$\}$ to be the set of event relation labels that an event pair $ep \in \mathcal{EP}$ can be associated with.

Note that the CEA metric as defined in Section 4.2.1 simply gives a score without it being bounded to be between 0 and 1.0. However, to use the CEA score as part of the inference process, we require that it be bounded and thus can be used as a binary prediction, that is, predicting an event pair as *causal* or ¬*causal*. To enable this, we use a few development documents to automatically find a threshold CEA score that separates scores indicating *causal* vs ¬*causal*. Based on this threshold, the original CEA scores are then rescaled to fall within 0 to 1.0. More details on this are in Section 4.6.2.

Let $\mathcal{C}$ denote the set of connective mentions in a document. We slightly modify our discourse system as follows. We define $\mathcal{L}_{DR}$ to be the set of discourse relations. We initially add all the fine-grained discourse relations listed in Table 4.1 to $\mathcal{L}_{DR}$. In the PDTB corpus, some connective examples are labeled with just a coarse-grained relation, without further specifying a fine-grained relation. To accommodate these examples, we add the coarse-grained relations *Comparison*, *Expansion*, and *Temporal* to $\mathcal{L}_{DR}$. We omit the coarse-grained *Contingency* relation from $\mathcal{L}_{DR}$, as we want to separate *Cause* and *Condition* from *Pragmatic-cause* and *Pragmatic-condition*. This discards very few examples as only a very small number of connective examples are simply labeled with a *Contingency* label without further specifying a fine-grained label. We then retrained our discourse system to predict labels in $\mathcal{L}_{DR}$.

4.5.2 Constraints

We now describe the constraints used to support joint inference, based on the predictions of the CEA metric and the discourse classifier. Let $s_c(dr)$ be the probability that connective c is predicated to be of discourse relation dr, based on the output of our discourse classifier. Let $s_{ep}(er)$ be the CEA prediction score (rescaled to range in [0,1]) that event pair ep takes on the *causal* or $\neg$*causal* label er. Let $x_{\langle c,dr\rangle}$ be a binary indicator variable which takes on the value 1 iff c is labeled with the discourse relation dr. Similarly, let $y_{\langle ep,er\rangle}$ be a binary variable which takes on the value 1 iff ep is labeled as er. We then define our objective function as follows:

$$\arg\max_{x,y}\Big[|\mathcal{L}_{DR}|\sum_{c\in\mathcal{C}}\sum_{dr\in\mathcal{L}_{DR}} s_c(dr)\cdot x_{\langle c,dr\rangle} + |\mathcal{L}_{ER}|\sum_{ep\in\mathcal{EP}}\sum_{er\in\mathcal{L}_{ER}} s_{ep}(er)\cdot y_{\langle ep,er\rangle}\Big] \tag{4.6}$$

subject to the following constraints:

$$\sum_{dr\in\mathcal{L}_{DR}} x_{\langle c,dr\rangle} = 1 \quad \forall c\in\mathcal{C} \tag{4.7}$$

$$\sum_{er\in\mathcal{L}_{ER}} y_{\langle ep,er\rangle} = 1 \quad \forall ep\in\mathcal{EP} \tag{4.8}$$

$$x_{\langle c,dr\rangle} \in \{0,1\} \quad \forall c\in\mathcal{C}, dr\in\mathcal{L}_{DR} \tag{4.9}$$

$$y_{\langle ep,er\rangle} \in \{0,1\} \quad \forall ep\in\mathcal{EP}, er\in\mathcal{L}_{ER} \tag{4.10}$$

Equation (4.7) requires that each connective c can only be assigned one discourse relation. Equation (4.8) requires that each event pair ep can only be *causal* or $\neg$*causal*. Equations (4.9) and (4.10) indicate that $x_{\langle c,dr\rangle}$ and $y_{\langle ep,er\rangle}$ are binary variables.

To capture the relationship between event pair causality and discourse relations, we use the following constraints:

$$x_{\langle c,\text{“}Cause\text{”}\rangle} \leq \sum_{ep\in\mathcal{EP}_c} y_{\langle ep,\text{“}causal\text{”}\rangle} \tag{4.11}$$

$$x_{\langle c,\text{“}Condition\text{”}\rangle} \leq \sum_{ep\in\mathcal{EP}_c} y_{\langle ep,\text{“}causal\text{”}\rangle}, \tag{4.12}$$

where both equations are defined $\forall c\in\mathcal{C}$. $\mathcal{EP}_c$ is defined to be the set of event pairs that cross the two text spans associated with c. For instance, if the first text span of c contains two event mentions

e_i, e_j, and there is one event mention e_k in the second text span of c, then $\mathcal{EP}_c = \{(e_i, e_k), (e_j, e_k)\}$. Finally, the logical form of Equation (4.11) can be written as: $x_{\langle c, \text{“}Cause\text{”}\rangle} \Rightarrow y_{\langle ep_i, \text{“}causal\text{”}\rangle} \vee \ldots \vee y_{\langle ep_j, \text{“}causal\text{”}\rangle}$, where $ep_i, \ldots, ep_j$ are elements in $\mathcal{EP}_c$. This states that if we assign the *Cause* discourse label to c, then at least one of $ep_i, \ldots, ep_j$ must be assigned as *causal*. The interpretation of Equation (4.12) is similar.

We use two more constraints to capture the interactions between event causality and discourse relations. First, we defined $\mathcal{C}_{ep}$ as the set of connectives c enclosing each event of ep in each of its text spans, i.e.: one of the text spans of c contain one of the event in ep, while the other text span of c contain the other event in ep. Next, based on the discourse relations in Section 4.4.1, we propose that when an event pair ep is judged to be *causal*, then the connective c that encloses it should be evoking one of the discourse relations in $\mathcal{L}_{DR_a} = \{$“Cause”, “Condition”, “Temporal”, “Asynchronous”, “Synchrony”, “Conjunction”$\}$. We capture this using the following constraint:

$$y_{\langle ep, \text{“}causal\text{”}\rangle} \leq \sum_{dr_a \in \mathcal{L}_{DR_a}} x_{\langle c, dr_a\rangle} \quad \forall c \in \mathcal{C}_{ep} \tag{4.13}$$

The logical form of Equation (4.13) can be written as:

$$y_{\langle ep, \text{“}causal\text{”}\rangle} \Rightarrow x_{\langle c, \text{“}Cause\text{”}\rangle} \vee x_{\langle c, \text{“}Condition\text{”}\rangle} \ldots \vee x_{\langle c, \text{“}Conjunction\text{”}\rangle}$$

This states that if we assign ep as *causal*, then we must assign to c one of the labels in $\mathcal{L}_{DR_a}$.

Finally, we propose that for any connectives evoking discourse relations $\mathcal{L}_{DR_b} = \{$“Comparison”, “Concession”, “Contrast”, “Pragmatic-concession”, “Pragmatic-contrast”, “Expansion”, “Alternative”, “Exception”, “Instantiation”, “List”, “Restatement”$\}$, any event pair(s) that it encloses should be $\neg$*causal*. We capture this using the following constraint:

$$\begin{aligned} x_{\langle c, dr_b\rangle} &\leq y_{\langle ep, \text{“}\neg causal\text{”}\rangle} \\ \forall\ dr_b \in \mathcal{L}_{DR_b},\ ep &\in \mathcal{EP}_c, \end{aligned} \tag{4.14}$$

where the logical form of Equation (4.14) can be written as: $x_{\langle c, dr_b\rangle} \Rightarrow y_{\langle ep, \text{“}\neg causal\text{”}\rangle}$.

4.6 Experiments

4.6.1 Experimental Settings

To collect the distributional statistics for measuring CEA as defined in Equation (4.1), we applied part-of-speech tagging, lemmatization, and dependency parsing [Marneffe et al., 2006] on about 760K documents in the English Gigaword corpus (LDC catalog number LDC2003T05).

We are not aware of any benchmark corpus for evaluating event causality extraction in contexts. Hence, we created an evaluation corpus using the following process: Using news articles collected from CNN[7] during the first three months of 2010, we randomly selected 20 articles (documents) as evaluation data, and 5 documents as development data.

Two annotators annotated the documents for causal event pairs, using two simple notions for causality: the Cause event should temporally precede the Effect event, and the Effect event occurs because the Cause event occurs. However, sometimes it is debatable whether two events are involved in a causal relation, or whether they are simply involved in an uninteresting temporal relation. Hence, we allowed annotations of C to indicate causality, and R to indicate relatedness (for situations when the existence of causality is debatable). The annotators will simply identify and annotate the C or R relations between predicates of event pairs. Event arguments are not explicitly annotated, although the annotators are free to look at the entire document text while making their annotation decisions. Finally, they are free to annotate relations between predicates that have any number of sentences in between and are not restricted to a fixed sentence window-size.

After adjudication, we obtained a total of 492 $C + R$ relation annotations, and 414 C relation annotations on the evaluation documents. On the development documents, we obtained 92 $C + R$ and 71 C relation annotations. The annotators overlapped on 10 evaluation documents. On these documents, the first (second) annotator annotated 215 (199) $C + R$ relations, agreeing on 166 of these relations. Together, they annotated 248 distinct relations. Using this number, their agreement ratio would be 0.67 (166/248). The corresponding agreement ratio for C relations is 0.58. These numbers highlight that causality identification is a difficult task, as there could be as many as N^2 event pairs in a document (N is the number of events in the document). We plan to make this

[7] http://www.cnn.com

System	Rec%	Pre%	F1%
PMI_{pp}	26.6	20.8	23.3
ECD_{pp} &$PMI_{pa,aa}$	40.9	23.5	29.9
CEA	62.2	28.0	38.6
CEA+Discourse	**65.1**	**30.7**	**41.7**

Table 4.2: Performance of baseline systems and our approaches on extracting *Causal* event relations.

System	Rec%	Pre%	F1%
PMI_{pp}	27.8	24.9	26.2
ECD_{pp} &$PMI_{pa,aa}$	42.4	28.5	34.1
CEA	63.1	33.7	43.9
CEA+Discourse	**65.3**	**36.5**	**46.9**

Table 4.3: Performance of the systems on extracting *Causal* and *Related* event relations.

annotated dataset available soon.[8]

4.6.2 Evaluation

As mentioned in Section 4.5.1, to enable translating (the unbounded) CEA scores into binary *causal*, ¬*causal* predictions, we need to rescale or calibrate these scores to range in [0,1]. To do this, we first rank all the CEA scores of all event pairs in the development documents. Most of these event pairs will be ¬*causal*. Based on the relation annotations in these development documents, we scanned through this ranked list of scores to locate the CEA score t that gives the highest F1-score (on the development documents) when used as a threshold between *causal* vs ¬*causal* decisions. We then ranked all the CEA scores of all event pairs gathered from the 760K Gigaword documents, discretized all scores higher than t into B bins, and all scores lower than t into B bins. Together, these $2B$ bins represent the range [0,1]. We used $B = 500$. Thus, consecutive bins represent a difference of 0.001 in calibrated scores.

To measure the causality between a pair of events e_i and e_j, a simple baseline is to calculate $PMI(p^i, p^j)$. Using a similar thresholding and calibration process to translate $PMI(p^i, p^j)$ scores into binary causality decisions, we obtained a F1 score of 23.1 when measured over the causality C relations, as shown in the row PMI_{pp} of Table 4.2.

As mentioned in Section 4.2.1, Riaz and Girju [Riaz and Girju, 2010] proposed the ECD metric to measure causality between two events. Thus, as a point of comparison, we replaced s_{pp} of

[8] http://cogcomp.cs.illinois.edu/page/publication_view/663

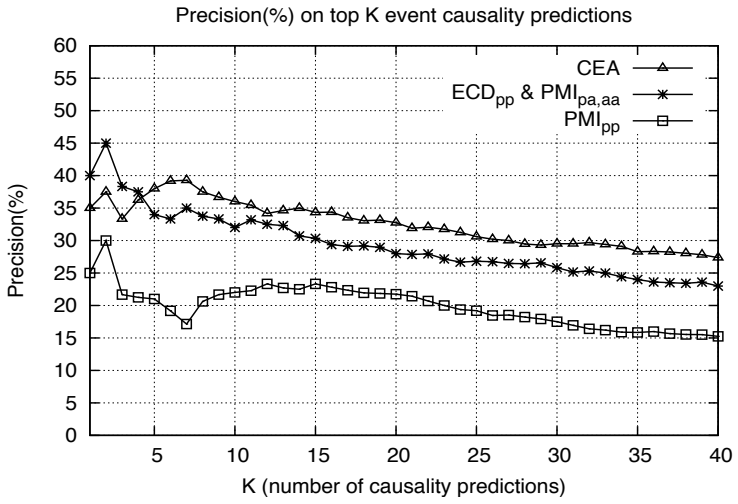

Figure 4.1: Precision of the top K causality C predictions.

Equation (4.1) with $ECD(a, b)$ of Equation (4.5), substituting $a = p^i$ and $b = p^j$. After thresholding and calibrating the scores of this approach, we obtained a F1-score of 29.7, as shown in the row $ECD_{pp}\&PMI_{pa,aa}$ of Table 4.2.

Next, we evaluated our proposed CEA approach and obtained a F1-score of 38.6, as shown in the row CEA of Table 4.2. Thus, our proposed approach obtained significantly better performance than the PMI baseline and the ECD approach. Next, we performed joint inference with the discourse relation predictions as described in Section 4.5 and obtained an improved F1-score of 41.7. We note that we obtained improvements in both recall and precision. This means that with the aid of discourse relations, we are able to recover more causal relations, as well as reduce false-positive predictions.

Constraint Equations (4.11) and (4.12) help to recover causal relations. For improvements in precision, as stated in the last paragraph of Section 4.5.2, identifying other discourse relations such as "Comparison", "Contrast", etc., provides counter-evidence to causality. Together with constraint Equation (4.14), this helps to eliminate false-positive event pairs as classified by CEA and contributes towards CEA+Discourse having a higher precision than CEA.

The corresponding results for extracting both causality and relatedness $C + R$ relations are given in Table 4.3. For these experiments, the aim was for a more relaxed evaluation and we simply collapsed C and R into a single label.

Finally, we also measured the precision of the top K causality C predictions, showing the precision trends in Figure 4.1. As shown, CEA in general achieves higher precision when compared to PMI_{pp} and $ECD_{pp}\&PMI_{pa,aa}$. The trends for $C + R$ predictions are similar.

Thus far, we had included both verbal and nominal predicates in our evaluation. When we repeat the experiments for $ECD_{pp}\&PMI_{pa,aa}$ and CEA on just verbal predicates, we obtained the respective F1-scores of 31.8 and 38.3 on causality relations. The corresponding F1-scores for casuality and relatedness relations are 35.7 and 43.3. These absolute F1-scores are similar to those in Tables 4.2 and 4.3, differing by 1-2%.

4.7 Analysis

We randomly selected 50 false-positive predictions and 50 false-negative *causality* relations to analyze the mistakes made by CEA.

Among the false-positives (precision errors), the most frequent error type (56% of the errors) is that CEA simply assigns a high score to event pairs that are not causal; more knowledge sources are required to support better predictions in these cases. The next largest group of error (22%) involves events containing pronouns (e.g. "he", "it") as arguments. Applying coreference to replace these pronouns with their canonical entity strings or labeling them with semantic class information might be useful.

Among the false-negatives (recall errors), 23% of the errors are due to CEA simply assigning a low score to causal event pairs and more contextual knowledge seems necessary for better predictions. 19% of the recall errors arises from causal event pairs involving nominal predicates that are not in our list of event evoking noun types (described in Section 4.3). A related 17% of recall errors involves nominal predicates without any argument. For these, less information is available for CEA to make predictions. The remaining group (15% of errors) involves events containing pronouns as arguments.

4.8 Related Work

Although prior work in event causality extraction in context is relatively sparse, there are many prior works concerning other semantic aspects of event extraction. In this section, we will give an overview on the related work of the event extraction and the event relation discovery tasks.

4.8.1 Event Extraction

Event extraction is the task of automatically identifying events in free text (e.g. news, emails and health report) and deriving detailed information about them. Informally, we want to extract the information about *who did what to whom, where and when*. In addition, events may also require to know some other pieces of information such as *what instrument/method was used*, *the reason of the event to happen*, etc.

There has been much work on event extraction in literature, especially after the introduction of the Automatic Content Extraction corpus in 2005 (ACE05)[9] which incorporated human annotation on events in the document collection. We are going to visit this corpus again soon, but for now, we will go over some work on automatic event extraction. We can roughly group the work in this field into two main groups based on their approaches including pattern-based event extraction and machine learning-based event extraction.

Some work using pattern-based approach includes [Grishman et al., 2002, Tanev et al., 2008]. Work in this group employs event patterns that will be used to match against potential events in text. Event patterns can be manually defined or semi-automatically acquired from text. In [Grishman et al., 2002], the authors describe a complete system that automatically extracts and updates an event database of infectious disease outbreaks. The system consists of a web crawler to retrieve news stories from multiple sources, a text zoner which parses input document into zones (e.g. headline, date, text body), and an extraction engine which actually does the work of extracting events from texts using predefined event patterns. The patterns include lists of event phrases (e.g. "outbreak of ...") and clauses (e.g. "people died from... ") that serve as event language markers. The system is designed to act in real-time situation, quickly analyze currently published news. On the other hand, [Tanev et al., 2008] also develop a system, named NEXUS,that works in real-time

[9] http://projects.ldc.upenn.edu/ace/

and employs syntactic patterns to extract events, but focuses on extracting news on violent and natural disasters. Especially, the patterns in this work are learned by a semi-automatic pattern acquisition technique consisting of a small corpus manually annotated with events and a pattern extraction procedure (we refer readers to [Tanev et al., 2008] for a more comprehensive description on the pattern acquisition procedure). Along with the learned patterns, NEXUS consists of several components chained in a pipeline starting from gathering news articles and ending at extracting events from the retrieved articles. Specifically, newly published article after gathered are clustered into news groups with documents on the same topic. Clusters on violent and natural disaster are selected and fed into an extraction engine that analyzes and retrieves events from all document in the clusters. The document are also preprocessed with natural language processing tools including tokenizer, sentence splitter, name entity recognizer, chunker, etc. The learned event patterns will be applied only after documents are preprocessed to extract events in the documents. However, for scaling purpose, NEXUS only analyzes and extracts events from the first sentence and the title of each document. This is because the authors assume that the most important events are usually expressed in the first sentence and the title of a document. Clearly, this assumption is arguable because it does not hold in several cases, but it simplifies the problem significantly.

On the other hand, work in machine learning-based approach mostly uses machine learning techniques to train one or more classifiers that can extract events in text at evaluation time. Some of the work in this direction also combines machine learning techniques with event patterns to precisely address the event extraction problem. Typical work in this direction includes [Hardy et al., 2006, Ahn, 2006, Bethard and Martin, 2006, Ji and Grishman, 2008]. In [Ahn, 2006], the authors propose a series of classification sub-tasks, each of which is handled by a machine-learned classifier. The system consists of an event anchor identifier, an argument identifier, an attribute assigner and an event coreference module. The anchor identification module finds event triggers which evokes events in text and assigns them an event type, while the argument identifier looks for the arguments of each event. Additionally, for each detected event, the attribute assignment module determines the values for the event's attributes including modality, polarity, genericity and tense. Finally, detected events that refer to the same event are discovered by the event coreference module. It is worth noting that all these modules are largely interdependent, but this fact is not taken into account when each

module is modeled. [Bethard and Martin, 2006] look at the problem in a different way and formulates event extraction as a sequence tagging problem with the **BIO** (Begin-Inside-Outside) schema (similar to models used to build shallow parsers [Munoz et al., 1999, Ramshaw and Marcus, 1995] . In this model, each word in text is assigned a label indicating whether it is **I**nside, **O**utside or **B**egin of an event. In this work, the authors further augment the B and I labels to capture the semantic class (a.k.a the type) of detected events. For example, there are eight semantic classes of events including I_ACTION, OCCURRENCE, etc., then the augmented labels include B_I_ACTION, B_OCCURRENCE, I_I_ACTION, I_OCCURRENCE, etc. Following this problem formulation, the authors focus on extracting learning feature for a single classifier that can be applied at the evaluation time to produce augmented BIO labels for words in text. The features are divided into eight groups, which include: text features, affix features, morphological features, word classes (including part-of-speech, syntactic chunks and word clusters), governing features from dependency parses, temporal information, negation and WordNet hypernym features. The classifier in this work is trained with Support Vector Machine (SVM) and evaluated on the TimeBank corpus [Pustejovsky et al., 2003b]. The results are promising. However, it is clear that the model proposed in this work is rather simple because its goal is simply detecting event mentions in text without recognizing their arguments, which are important for applications such as question answering and message understanding systems. [Ji and Grishman, 2008], on the other hand, address a more complete event extraction system which can identify event triggers (words or phrases that evoke events in text), recognize the event arguments of each trigger and also their role in the event. For instance, we want to identify the events in the text below (informally, we want to know who does what to whom, where and when).

> Iraq held elections on 3/07/2010, to elect a new Parliament and a prime minister. The slate led by Prime Minister Nuri Kamal al-Maliki trailed one led by a former interim leader, Ayad Allawi, by 89 seats to 91.

Their system is expected to extract events conveyed in the text, such as *holdsElections*(Prime Minister, Iraq, [Nuri Kamal al-Maliki, Ayad Allawi], 3/07/2010). The *holdsElection* event is defined by the following components:

- **Event Trigger:** *held elections*, is the language marker (a.k.a. the predicate) that notifies the existence of an election event in the text.
- **Arguments and Their Role:** *Prime Minister* is the position of the election, *Iraq* is the organization holding the event, and *Nuri Kamal al-Maliki* and *Ayad Allawi* are the candidates running for the election. Furthermore, *3/07/2010* describes the temporal information of the event.

In this work, the authors evaluate their system on the ACE corpus[10], which defines 8 coarse-grained event types with 33 fine-grained sub-types in total. Each event type is provided a general structure that defines the event name and the arguments and their role that the event can take. The authors improve a baseline within-sentence event tagger (which was a state-of-the-art system) [Grishman et al., 2005]. The baseline consists of the following components:

- Trigger Labeling: identifies event makers in text and classifier their event type.
- Argument Classifier: recognizes argument mentions of each trigger.
- Role Classifier: classifies argument roles.
- Reportable-Event Classifier: decides if a detected event candidate is an event.

The authors propose two key insights that significantly improve the baseline system: (i) one trigger sense per cluster – intuitively, this idea states that for a collection of topically-related documents, a particular verb has a high likelihood of expressing the same sense in any document; it, therefore, is likely to be trigger (or no trigger) consistently and represents the same event type (if it is a trigger), (ii) one argument role per cluster – this idea states that each entity plays the same argument role (or no role) for events with the same type in a collection of related documents. The baseline event extractor is then applied on both the test document and the retrieved related documents. A cross document inference module will take the outputs of the baseline and infer the correct trigger, event type and argument roles for detected event in the test document using the information of the events identified in the related documents. Experimental evaluations show that

[10]http://projects.ldc.upenn.edu/ace/

this model outperforms the baseline by 7.6 F-measure points in trigger labeling and 6 points in argument labeling, without consulting additional annotated data.

Recently, [Li et al., 2011] further improve the extraction performance by exploiting and incorporating background information networks through topic modeling methods. The key insights are still similar to those in [Ji and Grishman, 2008] but the system follows a different procedure to extract events in test documents. In this work, the system first forms a particular training set for each test document. A training set of a particular test document which is recognized to be in topic cluster i^{th} consists of a set of *postive* documents in the same topic cluster of the test document and a set of *negative* documents in other topic clusters. For each training set of a particular test document, the system in [Ji and Grishman, 2008] is applied to train and perform a cross document inference on the events extracted in the test document to get final event predictions. Evaluation shows that the method proposed in this paper achieves state-of-the-art performance on both English and Chinese texts. However, it is obvious that this method is not scalable in practice because it requires to form a training set and retrain the whole model for each test document.

About the ACE05 corpus, this is a well-know event annotation resource with 8 predefined coarse-grained event types including Life, Movement, Transaction, Business, Conflict, Contact, Personnel and Justice. These 8 event types are further divided into 33 subtypes such as Life-Injure, Life-Die, Conflict-Attack, Conflict-Demonstrate, etc. With 535 annotated documents, ACE05 provides a reasonable amount of data for training and evaluating event extraction systems with machine learning-based approaches. However, it is also worth noting that machine learning techniques usually limit the systems to some event types and not allow one to recognize unseen event types in future testing. Furthermore, data annotation is expensive and hard to obtain because of extensive labor work requirement. Recently, more and more work starts looking at unsupervised or semi-supervised approaches to address the event extraction problem. In addition, light-weight event extractors are another option to identifying events. For example, [Riaz and Girju, 2010] and our work described in this chapter develop simple approaches leveraging dependency parses to extract events. Their main focus was discovering event relations including causality and relatedness.

4.8.2 Event Relation Discovery

In this direction, previous work has focused on discovery multiple semantic relations between events. Below are some works that are related to our work described in this chapter.

Shahaf and Guestrin [Shahaf and Guestrin, 2010] investigated methods for automatically connecting topics and discovering hidden connections in this paper. They formalized the characteristics of a good chain and provided a linear program formulation to connect two endpoints. To formalize the story coherence they defined entities like (i) weak links, (ii) missing words, and (iii) jitteriness when analyzing a story and incorporating it to the formulation. The proposed LP formulation has $O(|D|^2 \times |W|)$ variables and is not feasible to scale up. The authors present some practical ways to speed up their formulation in the paper. Their formulation performed well in comparison to the Google News Timeline.

Chambers and Jurafsky [Chambers and Jurafsky, 2008b] introduce the concept of narrative event chains. The overall process has three steps:

- Learn narrative relations between events sharing coreferring arguments
- Apply a temporal classifier to partially order the connected events
- Prune and cluster self-contained chains from the space of events

Narrative chains are partially ordered sets of events centered on a common protagonist. Even though a narrative can have several participants, the assumption of the work is that there is a central character of the narrative (*the protagonist*). So, the narrative chains are structured by the protagonists grammatical roles in the events. A narrative chain is defined as a partially ordered set of narrative events that share a common actor. A *narrative event* is a tuple of an event (commonly a verb) and its participants represented by typed dependencies. So, here a narrative event is a tuple of the event and the typed dependency of the protagonist: *(event, dependency)*. Examples:

- (fired, ob) fired is the event and where the protagonist is the object in the sentence (the person being fired).
- (denied, subj) denied is the event and where the protagonist is the person being denied.

The first step is to learn narrative relations between events sharing coreferring arguments. The main approach for creating pair wise relations between events is to use a distributional score based on how often two events share grammatical arguments (using pointwise mutual information, PMI). That is, by counting pairs of verbs that share coreferring arguments within documents and computing the PMI between those verb-argument pairs. A global narrative score is built such that all events in the chain provide feedback on the given event.

The second step is to partially order the events in a narrative chain. This is achieved using a two-stage machine learning architecture. In the first stage, the model uses supervised learning to label temporal attributes of events. In the second stage, the classification from the first stage is used to classify the temporal relationship between two events. Note that in this work only the (strictly) BEFORE relation is considered.

Finally, the event space of narrative relations can be clustered to create discrete sets. This can be achieved by using the PMI scores from earlier in an agglomerative clustering algorithm and applying the ordering relations between events to produce a directed graph.

In another work, Chambers & Jurafsky [Chambers and Jurafsky, 2009] describes an approach to learn script-like information about the world, including both event structures and the roles of their participants, but without pre-defined roles or tagged corpora. This paper extends their earlier work [Chambers and Jurafsky, 2008b] to represent sets of situation-specific events similar to scripts. This work shows that verbs in distinct narrative chains can be merged into an improved single narrative schema, while the shared arguments across verbs can provide rich information for inducing semantic roles.

This paper extends the authors previous work by not restricting chains to a single protagonist and by including typed narrative chains that improves what can be inferred and learned from a corpus.

Riaz and Girzu [Riaz and Girju, 2010] presented an unsupervised approach to automatically identify contingency relationships between events in web news articles within and between sentences without relying on deep processing of contextual information. The approach focuses on a simple context consisting of two events, which if contingent, represents the cause (A) event and the effect event (B) such that A $\rightarrow$ B. Events are here considered to be [Subject] verb [Object]. The input to

the firsts step is a document corpus. The output from analysis is a set of contingency relationships.

The general approach follows three main steps:

1. Identifying topic-specific scenarios and their events
    (a) Discovering topic-specific scenarios
    (b) Identifying scenario-specific events
2. Generating event pair candidates
    (a) Grouping events
    (b) Identifying frequent event pairs
3. Learning contingency relations
    (a) Causal dependencies
    (b) Cause and effect roles assignments

4.9 Summary

In this work, using general tools such as the dependency and discourse parsers which are not trained specifically towards our target task, and a minimal set of development documents for threshold tuning, we developed a minimally supervised approach to identify causality relations between events in context. We also showed how to incorporate discourse relation predictions to aid event causality predictions through a global inference procedure. There are several interesting directions for future work, including the incorporation of other knowledge sources such as coreference and semantic class predictions, which were shown to be potentially important in our error analysis. We could also use discourse relations to aid in extracting other semantic relations between events.

Chapter 5

Event Timeline Construction

This work addresses the task of constructing a timeline of events mentioned in a given text. To accomplish that, we present a novel representation of the temporal structure of a news article based on time intervals. We then present an algorithmic approach that jointly optimizes the temporal structure by coupling local classifiers that predict associations and temporal relations between pairs of temporal entities with global constraints. Moreover, we present ways to leverage knowledge provided by event coreference to further improve the system performance. Overall, our experiments show that the joint inference model significantly outperformed the local classifiers by 9.2% of relative improvement in F_1. The experiments also suggest that good event coreference could make remarkable contribution to a robust event timeline construction system.

5.1 Introduction

Inferring temporal relations amongst a collection of events in a text is a significant step towards various important tasks such as automatic information extraction and document comprehension. Over the past few years, with the development of the TimeBank corpus [Pustejovsky et al., 2003b] , there have been several works on building automatic systems for such a task [Mani et al., 2006a, Chambers and Jurafsky, 2008a, Yoshikawa et al., 2009, Denis and Muller, 2011a].

Most previous works devoted much efforts to the task of identifying relative temporal relations (such as *before*, or *overlap*) amongst events [Chambers and Jurafsky, 2008a, Denis and Muller, 2011a], without addressing the task of identifying correct associations between events and their absolute time of occurrence. Even if this issue is addressed, certain restrictions are often imposed for efficiency reasons [Yoshikawa et al., 2009, Verhagen et al., 2010]. In practice, however, being able to automatically infer the correct time of occurrence associated with each event is crucial. Such

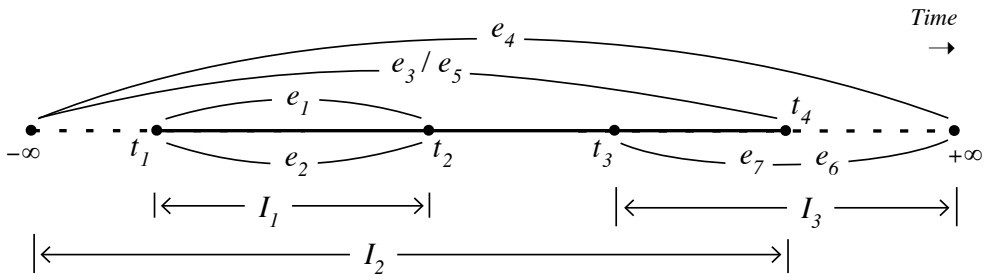

Figure 5.1: A graphical illustration of our timeline representation. The e's, t's and I's are events, time points and time intervals, respectively.

information not only leads to better text comprehension, but also enables fusion of event structures extracted from multiple articles or domains.

In this work, we are specifically interested in mapping events into an universal *timeline* representation. Besides inferring the relative temporal relations amongst the events, we would also like to automatically infer a specific absolute time of occurrence for each event mentioned in the text. Unlike previous work, we associate each event with a specific absolute time interval inferred from the text. An example timeline representation is illustrated in Fig. 5.1. We provide further details of our timeline representation in Sec. 5.3.3.

We perform global inference by combining a collection of local pairwise classifiers through the use of an Integer Linear Programming (ILP) formulation that promotes global coherence among local decisions. Our formulation allows our model to predict both event-event relation information and event-time interval association information simultaneously. We show that the approach of using event-time interval association not only produces desired results in a more concise way, but also reduces the number of variables and constraints required in the ILP.

Moreover, we observed that event coreference can reveal important information for such a task. We propose that different event mentions that refer to the same event can be grouped together before classifying and performing global inference. This can reduce the amount of efforts in both classification and inference stages and can potentially eliminate mistakes that would be made otherwise without such coreference information. To the best of our knowledge, our proposal of leveraging event coreference to support event timeline construction is novel.

Our experiments on a collection of annotated news articles from the standard ACE dataset demonstrate that it is possible to construct robust timelines of events with our approach. We

show that our algorithmic approach is able to combine various local evidences to produce a global coherent temporal structure, with improved overall performance. Furthermore, the experiments show that the overall performance can be further improved by exploiting knowledge from event coreference.

5.2 Related Work

In this section, we review work on temporal relation identification between events and ordering events based on their temporal information. It is well-known that research in this direction has got attention for a long time, dated back to 1980s and even before that. [Allen, 1983] introduce an interval-based temporal logic which is still being used today. In this work, the author represents times as intervals with starting and ending endpoints, (t, s), respectively. This representation allows the author to construct 5 interval relations defined by endpoints including less-than, equal-to, overlaps, meets and during. Further subdivinding the *during* relation (for a better computation model), the author obtains the following 7 base temporal interval relations: equal, before, meets, overlaps, during, starts, finishes. Together with their inverse relations (except *equal*), there are 13 temporal relations in total. To do reasoning on temporal information, in this work, the author also introduces the transitivity closure property among relations. For example if (A overlaps B) and (B before C), then (A before C).

Recently, the fact that annotation resources on temporal relations are made available, such as TimeBank [Pustejovsky et al., 2003b], has boosted the gains from work in this field by employing machine learning techniques. This corpus, later, was used (with some modifications and/or simplifications) in two TempEval challenges organized in 2007 and 2010. In this research direction, tasks defined include identifying temporal expressions in text, associating events to its occurrence time, recognizing temporal relations among events and temporal expressions including (events vs. tempexes), (tempexes vs. tempexes) and (events vs. events).

[Strötgen and Gertz, 2010a] address the problem of temporal expression identification. This work introduces HeidelTime, a system that achieved the best performance in the TempEval-2 challenge [Verhagen et al., 2010] and is still the state-of-the-art of this task. Interestingly, although systems employing machine learning techniques are very advanced and provide competitive per-

formance, HeidelTime is a rule-based system mainly using regular expression patterns to extract temporal expressions and knowledge resources as well as linguistic clues to normalize them. In the TempEval-2 challenge, HeidelTime reached 86.0 F-measure points (the best score) in the extraction task and the best results in assigning the correct value attributes.

On the other hand, [Mani et al., 2006b, Bethard et al., 2007a] investigate machine learning approaches to address the problem of ordering events in text. The main ideas in [Mani et al., 2006b] are actually expanding the training data derived from TimeBank using temporal reasoning and learning a *local* classifier that can labels temporal links among events and time expressions. The link labels follow 13 temporal relations in [Allen, 1983], but are simplified and reduced to 6 to be able to handled well, including ibefore, begins, ends, simultaneous, includes and before. The authors show that expanding training data by using transitivity closure property of temporal relations does help improving their system's performance significantly (from 62.5% to 93.1% for the 6-way classification). However, an obvious limitation of this *local* classifier is its inability of taking advantage of the algebraic properties (or reasoning) of temporal relations. It, therefore, does not maintain the consistency among the predicted relations at document level.

Recently, there has been much work attempting to leverage Allen's interval algebra of temporal relations [Allen, 1983] to enforce the global coherence of temporal relation predictions. As in [Tatu and Srikanth, 2008], the authors take advantage of Allen's algebra extensively not only to expand training data but also to find temporal inconsistencies and replace some of the relations contributing to the inconsistency by lower confident relations returned by the learned local classifier. This is actually a greedy search technique that approximately selects the best configuration of temporal relations among events and temporal expressions. It is worth noting that [Bramsen et al., 2006] also experiment with various other greedy inference schemes. For exact inferences, [Bramsen et al., 2006, Chambers and Jurafsky, 2008a] formulate the temporal reasoning problem in an integer linear programming (ILP) with constraints derived from the Allen's algebraic properties of temporal relations. However, these work restricts the set of temporal relations to only 3 labels: before, after and null (i.e. no relation). This makes a huge difference with other work that tackles a larger set of relations, and actually gives a clear advantage when performing temporal inference with interval algebra. Similarly, [Yoshikawa et al., 2009] also formulate the

problem as a global inference model, but employ Markov Logic Network (instead of ILP) to jointly predict temporal relations among events and temporal expressions in document. More recently, [Denis and Muller, 2011b] introduce an inference model, which subjects to constraints enforced on event endpoints, that allows to deal with more temporal relations without blowing up the constraint space. It is worth noting that these works are based on the temporal graph representation where temporal entities (i.e. temporal expressions and event mentions) are mounted on a graph as vertexes with the edges between them representing temporal relations.

In another type of temporal representation, [Amigo et al., 2011] introduce a 4-tuple representation that can represent temporal ranges for both the beginning and ending timepoints. In this representation, the temporal information of an event is denoted by a tuple $\langle t_1, t_2, t_3, t_4 \rangle$, where t_1 and t_2 are the lower and upper bounds of the beginning timepoint of the event and t_3 and t_4 are the lower and upper bounds of the ending. We believe that this representation is expressive, but may complicate the inference model given that there are more timepoints to enforce final output agreements.

5.3 Preliminaries

We focus on the task of mapping event mentions in a news article to a timeline. We first briefly describe and define several basic concepts.

5.3.1 Events

Following the annotation guidelines of the ACE project, we define an event as an action or occurrence that happens with associated participants or arguments. We also distinguish between events and event mentions, where a unique event can be coreferred to by a set of explicit event mentions in an article. Formally, an event E^i is co-referred to by a set of event mentions $(e_1^i, e_2^i, \ldots, e_k^i)$. Each event mention e can be written as $p(a_1, a_2, \ldots, a_l)$, where the predicate p is the word that triggers the presence of e in text, and $a_1, a_2, \ldots a_l$ are the arguments associated with e. In this work we focus on four temporal relations between two event mentions including *before*, *after*, *overlap* and *no relation*.

5.3.2 Time Intervals

Similar to [Denis and Muller, 2011a], we define time intervals as pairs of time endpoints. Each time interval I is denoted by $[t^-, t^+]$, where t^- and t^+ are two time endpoints representing the lower and upper bound of the interval I, respectively, with $t^- \leq t^+$. The general form of a time endpoint is written as "*YYYY-MM-DD hh:mm:ss*". An endpoint can be undefined, in which case it is set to an infinity value: $-\infty$, or $+\infty$. There are two types of time intervals:

Explicit intervals are time intervals that can be extracted directly from a given text. For example, consider the following snippet of an article in our data set: *The litigation covers buyers in auctions outside the United States between January 1, 1993 and February 7, 2000.* In this example, we can extract and normalize two time intervals which are explicitly written, including *January 1, 1993* → [1993-01-01 00:00:00, 1993-01-01 23:59:59] and *February 7, 2000* → [2000-02-07 00:00:00, 2000-02-07 23:59:59]. Moreover, an explicit interval can also be formed by one or more separate explicit temporal expressions. In the example above, the connective term *between* relates the two expressions to form another time interval: *between January 1, 1993 and February 7, 2000* → [1993-01-01 00:00:00, 2000-02-07 23:59:59]. To extract explicit time intervals from text, we use the time interval extractor described in Section 5.4.

Implicit intervals are time intervals that are not explicitly mentioned in the text. We observed that there are events that cannot be assigned to any precise time interval but are roughly known to occur in the past or in the future relative to the Document Creation Time (DCT) of the article. We introduce two implicit time intervals to represent the past and the future events as $(-\infty, t^-_{DCT}]$ and $[t^+_{DCT}, +\infty)$, respectively. In addition, we also allow an event mention to be assigned into the entire timeline, which is denoted by $(-\infty, +\infty)$ if we cannot identify its time of occurrence. We also consider DCT as an implicit interval.

We say that the time interval I_i precedes the time interval I_j on a timeline if and only if $t^+_i \leq t^-_j$, which also implies that I_i succeeds I_j if and only if $t^-_i \geq t^+_j$. The two intervals overlap, otherwise.

5.3.3 Timeline

We define a timeline as a partially ordered set of time intervals. Fig. 5.1 gives a graphical illustration of an example timeline, where events are annotated and associated with time intervals. Relations

amongst events can be properly reflected in the timeline representation. For example, in the figure, the events e_1 and e_2 are both associated with the interval I_1. The relation between them is *no relation*, since it is unclear which occurs first. On the other hand, e_5 and e_3 both happen in the interval I_2 but they form an *overlap* relation. The events e_6 and e_7 occur within the same interval I_3, but e_7 precedes (i.e. *before*) e_6 on the timeline. The event e_4 is associated with the interval $(-\infty, +\infty)$, indicating there is no knowledge about its time period of occurrence.

We believe that such a timeline representation for temporally ordering events has several advantages over the temporal graph representations in previous works [Chambers and Jurafsky, 2008a, Yoshikawa et al., 2009, Denis and Muller, 2011a]. Unlike previous works, in our model the events are partially ordered in a single timeline, where each event is associated with a precise time interval. This improves human interpretability of the temporal relations amongst events and time. This property of our timeline representation, thus, facilitates merging multiple timelines induced from different articles. Furthermore, as we will show later, the use of time intervals within the timeline representation simplifies the global inference formulation and thus the inference process.

5.4 Fundamental Time Interval Operations

Performing temporal reasoning with respect to temporal expressions is important in many NLP tasks such as text summarization, information extraction, discourse understanding and information retrieval. Recently, the Knowledge Base Population track [Ji et al., 2011] introduced the temporal slot filling task that requires identifying and extracting temporal information for a limited set of binary relations such as (*person*, *employee_of*), (*person*, *spouse*). In the work of [Wang et al., 2010], the authors presented the Timely Yago ontology, which extracted and incorporated temporal information as part of the description of the events and relations in the ontology. Temporal reasoning is also essential in supporting the emerging temporal information retrieval research direction [Alonso et al., 2011].

In this section, we present a system that addresses three fundamental tasks in temporal reasoning:

Extraction: Capturing the extent of time expressions in a given text. This task is based on task A in the TempEval-2 challenge [Verhagen et al., 2010]. Consider the following sentence:

Seventy-five million copies of the rifle have been built since it entered production in February 1947.

In this sentence, *February 1947* is a basic temporal expression that should be extracted by the extraction module. More importantly, we further extend the task to support also the extraction of *complex temporal expressions* that are not addressed by existing systems. In the example above, it is important to recognize and capture the phrase *since it entered production in February 1947* as another temporal expression that expresses the time period of the *manufacturing* event (triggered by *built*.) For the best of our knowledge, this extension is novel.

Normalization: Normalizing temporal expressions, which are extracted by the extraction module, to a canonical form. Our system normalizes temporal expressions (including complex ones) to time intervals of the form [*start point*, *end point*]. The endpoints follow a standard date and time format: *YYYY-MM-DD hh:mm:ss*. Our system accounts for an input reference date when performing the normalization. For example, given March 20^{th}, 1947 as a reference date, our system normalizes the temporal expressions extracted in the example above as follows: [*1947-02-01 00:00:00*, *1947-02-28 23:59:59*] and [*1947-02-01 00:00:00*, *1947-03-20 23:59:59*], respectively.

Comparison: Comparing two time intervals (i.e. normalized temporal expressions). This module identifies the temporal relation that holds between intervals, including the *before*, *before-and-overlap*, *containing*, *equal*, *inside* , *after* and *after-and-overlap* relations. For example, when comparing the two normalized time intervals above, we get the following result: [*1947-02-01 00:00:00*, *1947-02-28 23:59:59*] is *inside* [*1947-02-01 00:00:00*, *1947-03-20 23:59:59*].

There has been much work addressing the problems of temporal expression extraction and normalization, i.e. the systems developed in TempEval-2 challenge [Verhagen et al., 2010]. However, our system is different from them in several aspects. First, we extend the extraction task to capture complex temporal expressions. Second, our system normalizes temporal expressions (including complex ones) to time intervals instead of time points. Finally, our system performs temporal comparison of time intervals with respect to multiple relations. We believe that with the rapid progress in NLP and IR, more tasks will require temporal information and reasoning, and a system that addresses these three fundamental tasks well will be able to support and facilitate temporal reasoning systems efficiently.

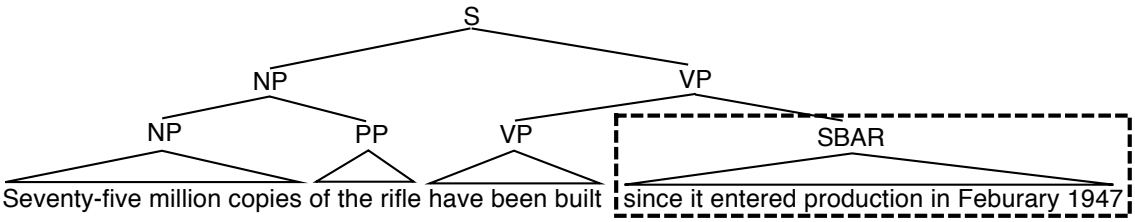

Figure 5.2: The SBAR constituent in the parse tree determines an extended temporal expression given that *in February 1947* is already captured by HeidelTime.

5.4.1 Temporal Expression Extraction

We built the temporal expression extraction module on top of the Heideltime system, which was described in [Strötgen and Gertz, 2010b], to take advantage of a state-of-the-art temporal extraction system in capturing basic expressions. We use the Illinois POS tagger[1] [Roth and Zelenko, 1998] to provide part-of-speech tags for the input text before passing it to HeidelTime. Below is an example of the HeidelTime output of the example in the previous section:

> Seventy-five million copies of the rifle have been built since it entered production in <TIMEX3 tid="t2" type="DATE" value="1947-02">February 1947</TIMEX3>

In this example, HeidelTime captures a basic temporal expression: *February 1947*. However, HeidelTime cannot capture the complex temporal expression *since it entered production in February 1947*, which expresses a period of time from February 1947 until the document creation time. This is actually the time period of the manufacturing event (triggered by *built*). To capture complex phrases, we make use of a syntactic parse tree[2] as illustrated in Figure 5.2. A complex temporal expression is recognized if it satisfies the following conditions:

- It is covered by a PP or SBAR constituent in the parse tree.
- The constituent starts with a temporal connective. In this work, we focus on an important subset of temporal connectives, consisting of *since*, *between*, *from*, *before* and *after*.
- It contains at least one basic temporal expression extracted by HeidelTime.

[1] http://cogcomp.cs.illinois.edu/page/software_view/POS
[2] We use nlparser [Charniak and Johnson, 2005]

In addition, our extraction module also handles holidays in several countries. For example, in the sentence "The gas price increased rapidly after Christmas.", we are able to extract two temporal expressions *Christmas* and *after Christmas*, which refer to different time intervals.

5.4.2 Normalization to Time Intervals

Our system normalizes a temporal expression to a time interval of the form [*start point*, *end point*], where *start point* $\leq$ *end point*. Each time endpoint of an interval follows a standard date and time format: *YYYY-MM-DD hh:mm:ss*. It is worth noting that this format augments the date format in TimeML, used by HeidelTime and other existing systems. Our date and time format of each time endpoint refer to an absolute time point on a universal timeline, making our time intervals absolute as well. Furthermore, we take advantage of the predicted temporal value of each temporal expression from the HeidelTime output. For instance, in the HeidelTime output example above, we extract *1947-02* as the normalized date of *February 1947* and then convert it to the interval [*1947-02-01 00:00:00*, *1947-02-28 23:59:59*]. If HeidelTime cannot identify an exact date, month or year, we then resort to our own temporal normalizer, which consists of a set of conversion rules, regarding to the document creation time of the input text. An interval endpoint can get infinity value if its temporal boundary cannot be specified.

5.4.3 Comparison

To compare two time intervals (i.e. normalized temporal expressions), we define six temporal relations: *before*, *before-and-overlap*, *contains*, *equals*, *inside*, *after* and *after-and-overlap*. The temporal relation between two normalized intervals is determined by a set of comparison rules that take the four interval endpoints into consideration. For example, $A = [s_A, e_A]$ *contains* $B = [s_B, e_B]$ if and only if $(s_A < s_B) \wedge (e_A > e_B)$, where s and e are intervals start and end points, respectively.

5.4.4 Experimental Study

In this section, we present an evaluation of our extended temporal extractor, the normalizer and the comparator. We do not evaluate the HeidelTime temporal extractor again because its performance was reported in the TempEval-2 challenge [Verhagen et al., 2010], where it achieved 0.86 F_1 score

Connective	# sent.	# appear.	Prec	Rec	F_1
since	31	31	1.0	1.0	1.0
between	32	33	1.0	1.0	1.0
from	340	366	0.8	1.0	0.89
before	33	33	0.8	1.0	0.89
after	78	81	0.72	1.0	0.84
Avg.			0.86	1.0	0.92

Table 5.1: The performance of our extended temporal extractor on complex expressions which contain at least one of the connectives shown in the first column. These expressions cannot be identified by existing systems.

on the TimeBank data sets [Pustejovsky et al., 2003a].

Data Preparation

We focus on scaling up temporal systems to deal with complex expressions. Therefore, we prepared an evaluation data set that consists of a list of sentences containing at least one of the five temporal connectives *since*, *betwen*, *from*, *before* and *after*. To do this, we extract all sentences that satisfy the condition from 183 articles in the TimeBank 1.2 corpus[3]. This results in a total of 486 sentences. Each sentence in the data set comes with the document creation time (DCT) of its corresponding article. The second and the third columns of Table 5.1 summarize the number of sentences and appearances of each temporal connective.

We use this data set to evaluate the extended temporal extractor, the normalizer and also the comparator of our system. We note that although this data set is driven by our focused temporal connectives, it does not lose the generality of evaluating the normalization and comparison modules because the sentences in this data set also contain many basic temporal expressions. Moreover, there are many cases where the connectives in our data are not actually temporal connectives. Our system is supposed to not capture them as temporal expressions. This is also reflected in the experimental results.

Experimental Results

We report the performance of our extended temporal extraction module using precision, recall and F_1 score as shown in the last three columns of Table 5.1. We evaluate the normalization module on

[3]http://www.ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId=LDC2006T08

Module	Correct	Incorrect	Acc.
Normalizer	191	16	0.92
Comparator	191	0	1.0

Table 5.2: The performance of the normalization and comparison modules. We only compare the 191 correctly identified time intervals with their corresponding document creation time.

the correctly extracted temporal expressions, including basic expressions captured by HeidelTime and the extended expressions identified by our extractor. A normalization is correct if and only if both time interval endpoints are correctly identified[4]. We study the comparison module by evaluating it on the comparisons of the correctly normalized expressions against the corresponding DCT of the sentences from which they are extracted. Because the normalization and comparison outputs are judged as correct or incorrect, we report the performance of these modules in accuracy (Acc) as shown in Table 5.2. Overall, the experimental study shows that all modules in our system are robust and achieve excellent performance.

5.5 A Joint Timeline Model

Our task is to induce a globally coherent timeline for a given article. We thus adopt a global inference model for performing the task. The model consists of two components: (1) two local pairwise classifiers, one between event mentions and time intervals (the *E–T* classifier) and one between event mentions themselves (the *E–E* classifier), and (2) a joint inference module that enforces global coherency constraints on the final outputs of the two local classifiers. Fig. 5.3 shows a simplified temporal structure of event mentions and time intervals of an article in our model.

Our *E–T* classifier is different from those in previous work [Chambers and Jurafsky, 2008a, Yoshikawa et al., 2009, Denis and Muller, 2011a] where such classifiers were trained to identify temporal relations between event mentions and a temporal expression. In our work, in order to construct absolute timeline of event mentions, temporal expressions are captured and normalized as absolute time intervals. The *E–T* classifiers are then used to assign event mentions to their contextually corresponding time intervals.

[4]We use hours as the level of granularity in our work.

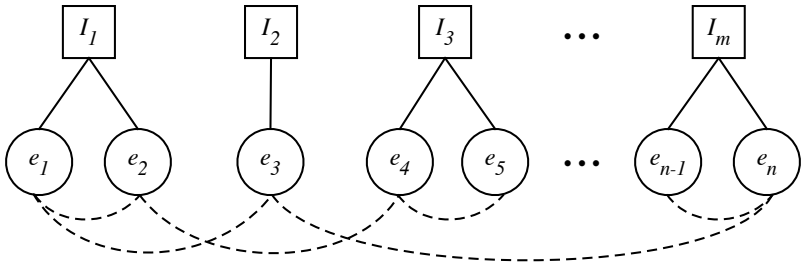

Figure 5.3: A simplified temporal structure of an article. There are m time intervals $I_1 \cdots I_m$ and n event mentions $e_1 \cdots e_n$. A solid edge indicates an association between an interval and an event mention, whereas a dash edge illustrates a temporal relation between two event mentions.

We also lifted many restrictions imposed in previous work, as described in [Bethard et al., 2007b, Yoshikawa et al., 2009, Verhagen et al., 2010]. For example, we do not require that event mentions and time expressions have to appear in the same sentence. Similarly, we do not require that event mentions have to appear very close to each other (e.g. main event mentions in adjacent sentences) are considered as candidate pairs for classification. In contrast, we performed classifications over all pairs of event mentions and time intervals as well as over all pairs of event mentions. Our experimental study showed that lifted these restrictions is indeed important (see Section 5.7).

Another important improvement over previous work is our global inference model We would like to highlight that our work is also distinct from most previous works in the global inference component. Specifically, our global inference model jointly optimizes the *E-E* relations amongst event mentions and their associations, *E-T*, with temporal information (intervals in our case). Previous work [Chambers and Jurafsky, 2008a, Denis and Muller, 2011a], on the other hand assumed that the *E-T* information is given and only tried to improve *E-E*.

5.5.1 The Pairwise Classifiers

We first describe our local classifiers that associate event mention with time interval and classify temporal relations between event mentions, respectively.

C_{E-T}: is the *E–T* classifier that associates an event mention with a time interval. Given an event mention and a time interval, the classifier predicts whether the former associates with the latter.

$$C_{E-T}(e_i, I_j) \rightarrow \{0, 1\},$$
$$\forall i, j, 1 \leq i \leq n, 1 \leq j \leq m, \tag{5.1}$$

where n and m are the number of event mentions and time intervals in an article, respectively.

C_{E-E}: is the E–E classifier that identifies the temporal relation between two event mentions. Given a pair of event mentions, the classifier predicts one of the four temporal relations between them: $\bar{b}$*efore*, $\bar{a}$*fter*, $\bar{o}$*verlap* and $\bar{n}o$ *relation*. Specifically:

$$C_{E-E}(e_i, e_j) \rightarrow \{\bar{b}, \bar{a}, \bar{o}, \bar{n}\},$$
$$\forall i, j, 1 \leq i, j \leq n, i \neq j, \tag{5.2}$$

For training of the classifiers, we define a set of features following some previous work such as [Bethard et al., 2007b, Chambers and Jurafsky, 2008a, Yoshikawa et al., 2009], together with some additional features that we believe to be helpful for the interval-based representation. We describe the base features below and use † and ‡ to denote the features used for C_{E-T} and C_{E-E}, respectively. We use the term *temporal entity* (or *entity*, for short) to refer to either an event mention or a time interval.

Lexical Features: A set of lexical features related to the temporal entities: (i)†‡ the word, lemma and part-of-speech of the input event mentions and the context surrounding them, where the context is defined as a window of 2 words before and after the mention; (ii)† the modal verbs to the left and to the right of the event mention; (iii)‡ the temporal connectives between the event mentions[5].

Syntactic Features: (i)†‡ which entity appears first in the text; (ii)†‡ whether the two entities appear in the same sentence; (iii)†‡ the quantized number of sentences between the two entities; (iv)†‡ whether the input event mentions are covered by prepositional phrases and what are the heads of the phrases; (v)†‡ if the entities are in the same sentence, what is their least common constituent on the syntactic parse tree; (vi)† whether there is any other temporal entity that is closer to one of the two entities.

Semantic Features‡: A set of semantic features, mostly related to the input event mentions:

[5]We define a list of temporal connectives including *before*, *after*, *since*, *when*, *meanwhile*, *lately*, etc.

(i) whether the input event mentions have a common synonym from their synsets in WordNet [Fellbaum, 1998]; (ii) whether the input event mentions have a common derivational form derived from WordNet.

Linguistic Features[†‡]: The tense and the aspect of the input event mentions. We use an in-house rule-based recognizer to extract these features.

Time Interval Features[†]: A set of features related to the input time interval: (i) whether the interval is implicit; (ii) if it is implicit, identify its interval type: "dct" $= [t^-_{DCT}, t^+_{DCT}]$, "past" $= (-\infty, t^-_{DCT}]$, "feature" $= [t^+_{DCT}, +\infty)$, and "entire" $= (-\infty, +\infty)$; (iii) the interval is before, after or overlapping with the DCT.

We note that unlike many previous work [Mani et al., 2006a, Chambers and Jurafsky, 2008a, Denis and Muller, 2011a], our classifiers do not use any gold annotations of event attributes (event class, tense, aspect, modal and polarity) provided in the TimeBank corpus as features.

In our work, we use a regularized averaged Perceptron [Freund and Schapire, 1999] as our classification algorithm[6]. We used the one-vs.-all scheme to transform a set of binary classifiers into a multi-class classifier (for C_{E-E}). The raw prediction scores were converted into probability distribution using the Softmax function (Bishop 1996). If there are n classes and the raw score of class i is act_i, the posterior estimation for class i is:

$$\tilde{P}(i) = \frac{e^{act_i}}{\sum_{1 \le j \le n} e^{act_j}}$$

5.5.2 Joint Inference for Event Timeline

To exploit the interaction among the temporal entities in an article, we optimize the predicted temporal structure, formed by predictions from C_{E-T} and C_{E-E}, w.r.t. a set of global constraints that enforce coherency on the final structure. We perform exact inference using Integer Linear Programming (ILP) as in [Roth and Yih, 2007, Clarke and Lapata, 2008]. We use the Gurobi Optimizer[7] as a solver.

Let $\mathcal{I} = \{I_1, I_2, \ldots, I_m\}$ denote the set of time intervals extracted from an article, and let $\mathcal{E} = \{e_1, e_2, \ldots, e_n\}$ denote all event mentions in the same article. Let $\mathcal{EI} = \{(e_i, I_j) \in \mathcal{E} \times \mathcal{I} | e_i \in$

[6]Other algorithm (e.g. SVM) gave comparable or worse results, so we only show the results from Averaged Perceptron.

[7]http://gurobi.com/

$\mathcal{E}, I_j \in \mathcal{I}\}$ denote the set of all pairs of event mentions and time intervals. We also denote the set of event mention pairs by $\mathcal{EE} = \{(e_i, e_j) \in \mathcal{E} \times \mathcal{E} | e_i \in \mathcal{E}, e_j \in \mathcal{E}, i \neq j\}$. The prediction probability of an association of a pair $eI \in \mathcal{EI}$, given by C_{E-T}, is denoted by $p_{\langle eI,1\rangle}$[8]. Now, let $\mathcal{R} = \{\bar{b}, \bar{a}, \bar{o}, \bar{n}\}$ be the set of temporal relations between two event mentions. The prediction probability of an event mention pair $ee \in \mathcal{EE}$ that takes temporal relation r, given by C_{E-E}, is denoted by $p_{\langle ee,r\rangle}$. Furthermore, we define $x_{\langle eI,1\rangle}$ to be a binary indicator variable that takes on the value 1 iff an association is predicted between e and I. Similarly, we define a binary indicator variable $y_{\langle ee,r\rangle}$ of a pair of event mentions ee that takes on the value 1 iff ee is predicted to hold the relation r. We then define our objective function as follows:

$$\arg\max_{x,y} \Big[\lambda \sum_{eI \in \mathcal{EI}} p_{\langle eI,1\rangle} \cdot x_{\langle eI,1\rangle} + (1-\lambda) \sum_{ee \in \mathcal{EE}} \sum_{r \in \mathcal{R}} p_{\langle ee,r\rangle} \cdot y_{\langle ee,r\rangle}\Big] \tag{5.3}$$

subject to the following constraints:

$$x_{\langle eI,1\rangle} \in \{0,1\}, \qquad \forall eI \in \mathcal{EI} \tag{5.4}$$

$$y_{\langle ee,r\rangle} \in \{0,1\}, \qquad \forall ee \in \mathcal{EE}, r \in \mathcal{R} \tag{5.5}$$

$$\textstyle\sum_{r \in \mathcal{R}} y_{\langle ee,r\rangle} = 1, \qquad \forall ee \in \mathcal{EE} \tag{5.6}$$

We use the single parameter λ to balance the overall contribution of two components E-T and E-E. λ is determined through cross validation tuning on a development set. We use (5.4) and (5.5) to make sure $x_{\langle eI,1\rangle}$ and $y_{\langle ee,r\rangle}$ are binary values. The equality constraint (5.6) ensures that exactly one particular relation can be assigned to each event mention pair.

In addition, we also require that each event is associated with only one time interval. These constraints are encoded as follows:

$$\sum_{I \in \mathcal{I}} x_{\langle eI,1\rangle} = 1, \qquad \forall e \in \mathcal{E} \tag{5.7}$$

Our model also enforces reflexivity and transitivity constraints on the relations among event mentions as follows:

[8]This value is complementary to the non-association probability, denoted by $p_{\langle eI,0\rangle} = 1 - p_{\langle eI,1\rangle}$

$$y_{\langle e_i e_j, r\rangle} - y_{\langle e_j e_i, \hat{r}\rangle} = 0,$$
$$\forall e_i e_j = (e_i, e_j) \in \mathcal{EE}, i \neq j \tag{5.8}$$

$$y_{\langle e_i e_j, r_1\rangle} + y_{\langle e_j e_k, r_2\rangle} - y_{\langle e_i e_k, r_3\rangle} \leq 1,$$
$$\forall e_i e_j, e_j e_k, e_i e_k \in \mathcal{EE}, i \neq j \neq k \tag{5.9}$$

The equality constraints in (5.8) encode reflexive property of event-event relations, where the relation $\hat{r}$ denotes the inversion of the relation r. The set of possible $(r, \hat{r})$ pairs is defined as follows: $\big\{(\bar{b}, \bar{a}), (\bar{a}, \bar{b}), (\bar{o}, \bar{o}), (\bar{n}, \bar{n})\big\}$. Following the work described in [Bramsen et al., 2006, Chambers and Jurafsky, 2008a], we encode transitive closure of relations between event mentions with inequality constraints in (5.9), which states that if the pair (e_i, e_j) has a certain relation r_1, and the pair (e_j, e_k) has the relation r_2, then the relation r_3 must be satisfied between e_i and e_k. Examples of such triple (r_1, r_2, r_3) include ($\bar{b}$, $\bar{b}$, $\bar{b}$) and ($\bar{a}$, $\bar{a}$, $\bar{a}$).

Finally, to capture the interactions between our local pairwise classifiers we add the following constraints:

$$x_{\langle e_i I_k, 1\rangle} + x_{\langle e_j I_l, 1\rangle} - y_{\langle e_i e_j, \bar{b}\rangle} \leq 1,$$
$$\forall e_i I_k, e_j I_l \in \mathcal{EI}, \forall e_i e_j \in \mathcal{EE},$$
$$I_k \text{ precedes } I_l, i \neq j, k \neq l \tag{5.10}$$

Intuitively, the inequality constraints in (5.10) specify that a temporal relation between two event mentions can be inferred from their respective associated time intervals. Specifically, if two event mentions e_i and e_j are associated with two time intervals I_k and I_l respectively, and I_k precedes I_l in the timeline, then e_i must happen before e_j.

It is important to note that our interval-based formulation is more concise in terms of the numbers variables and constraints imposed in the ILP over the formulations based on time expression (or timepoint) used in previous work [Chambers and Jurafsky, 2008a]. Specifically, in such timepoint-based formulations, the relation between each event mention and each time expression needs to be inferred, resulting in $|\mathcal{E}||\mathcal{T}||\mathcal{R}_\mathcal{T}|$ variables, where $|\mathcal{E}|$, $|\mathcal{T}|$, and $|\mathcal{R}_\mathcal{T}|$ are the numbers of event mentions, time points, and temporal relations respectively. In contrast, only $|\mathcal{E}||\mathcal{I}|$ variables

are required in our formulation, where $|\mathcal{I}|$ is the number of intervals (since we extract intervals explicitly, $|\mathcal{I}|$ is roughly equal to $|\mathcal{T}|$). Furthermore, performing inference with the timepoint-based formulation would require $|\mathcal{E}||\mathcal{T}|$ equality constraints to enforce that each event mention can take only one relation in $\mathcal{R}_{\mathcal{T}}$ to a particular time point, whereas our interval-based model only requires $|\mathcal{E}|$ constraints, since each event is strictly associated with one interval (see Eqn. (5.7)). We justify the benefits of our formulation later in Sec. 5.7.4.

5.6 Incorporating Knowledge from Event Coreference

One of the key contributions of our work is using event coreference information to enhance the timeline construction performance. This is motivated by the following two principles:

(P1) *All mentions of a unique event are associated with the same time interval, and overlap with each other.*

(P2) *All mentions of an event get the same temporal relation with all mentions of another event.*

The example below, extracted from an article published on 03/11/2003 in the Automatic Content Extraction (ACE), 2005, corpus[9] serves to illustrate the significance of event coreference to our task.

> *The world's most powerful fine art auction houses, Sotheby's and Christie's, have agreed to* $[e_1^1 =$ **pay**$]$ *40 million dollars to settle an international price-fixing scam, Sotheby's said. The* $[e_2^1 =$ **payment**$]$*, if approved by the courts, would settle a slew of* $[e_1^2 =$ **suits**$]$ *by clients over auctions held between 1993 and 2000 outside the US. ... Sotheby's and Christie's will each* $[e_3^1 =$ **pay**$]$ *20 million dollars," said Sotheby's, which operates in 34 countries.*

In this example, there are 4 event mentions, whose trigger words are highlighted in bold face. The underlined text gives an explicit time interval: I_1 = [1993-01-01 00:00:00, 2000-12-31 23:59:59] (we ignore 2 other intervals given by *1993* and *2000* to simplify the illustration). Now if we consider the event mention e_2^1, it actually belongs to the implicit future interval I_2 = [2003-03-11 23:59:59,

[9] http://www.itl.nist.gov/iad/mig/tests/ace/2005/

$+\infty$). Nevertheless, there is a reasonable chance that C_{E-T} associates it with I_1, given that they both appear in the same sentence, and there is no direct evident feature indicating the event will actually happen in the future. In such a situation, using a local classifier to identify the correct temporal association could be challenging.

Fortunately, precise knowledge from event coreference may help alleviate such a problem. The knowledge reveals that the 4 event mentions can be grouped into 2 distinct events: $E^1 = \{e_1^1, e_2^1, e_3^1\}$, $E^2 = \{e_1^2\}$. If C_{E-T} can make a strong prediction in associating the event mention e_1^1 (or e_3^1) to I_2, instead of I_1, the system will have a high chance to re-assign e_2^1 to I_2 based on principle (P1). Similarly, if C_{E-E} is effective in figuring out that some mention of event E^1 occurs *after* some mention of E^2, then all the mentions of E^1 would be predicted to occur *after* all mentions in E^2 according to (P2).

To incorporate knowledge from event coreference into our classifiers and the joint inference model, we use the following procedure: (1) performing classification with C_{E-T} and C_{E-E} on the data, (2) using the knowledge from event coreference to overwrite the prediction probabilities obtained by the two local classifiers in step (1), and (3) applying the joint inference model on the new prediction probabilities obtained from (2). We note that if we stop at step (2), we get the outputs of the local classifiers enhanced by event coreference knowledge.

To overwrite the classification probabilities using event coreference knowledge, we propose two approaches as follows:

MaxScore:

We define the probability between any mention $e \in E^i$ and an interval I as follows:

$$p_{\langle eI,1\rangle} = \max_{e' \in E^i} \tilde{P}(e', I) \tag{5.11}$$

where $\tilde{P}(e', I)$ is the classifier (C_{E-T}) probability for associating event mention e' to the time interval.

On the other hand, the probabilities for associating the set of temporal relations, $\mathcal{R}$, to each pair of mentions in $E^i \times E^j$, is given by the following pair:

$$\begin{aligned}(e^i, e^j)^* &= \underset{(e^{i'}, e^{j'}) \in E^i \times E^j, r \in \mathcal{R}}{\arg\max} \tilde{P}\big((e^{i'}, e^{j'}), r)\big) \\ p_{\langle ee, r\rangle} &= \tilde{P}\big((e^i, e^j)^*, r\big), \forall r \in \mathcal{R} \end{aligned} \tag{5.12}$$

In other words, over all possible event mention pairs and relations, we first pick the pair who globally obtains the highest probability for some relation. Next, we simply take the probability distribution of that event mention pair as the distribution over the relations, for the event pair.

SumScore: The probability between any mention $e \in E^i$ and an interval I is obtained by:

$$p_{\langle eI, 1\rangle} = \frac{1}{|E^i|} \sum_{e' \in E^i} \tilde{P}(e', I) \tag{5.13}$$

To obtain the probability distribution over the set of temporal relations, $\mathcal{R}$, for any pair of mentions in $E^i \times E^j$, we used the following procedure:

$$\begin{aligned} r^* &= \underset{r \in \mathcal{R}}{\arg\max} \sum_{e^i \in E^i} \sum_{e^j \in E^j} \tilde{P}\big((e^i, e^j), r\big) \\ (e^i, e^j)^* &= \underset{(e^{i'}, e^{j'}) \in E^i \times E^j}{\arg\max} \tilde{P}\big((e^{i'}, e^{j'}), r^*\big) \\ p_{\langle ee, r\rangle} &= \tilde{P}\big((e^i, e^j)^*, r\big), \forall r \in \mathcal{R} \end{aligned} \tag{5.14}$$

In other words, given two groups of event mentions, we first compute the total score of each relation, and select the relation which has the highest score. Next from the list of pairs of event mentions from the two groups, we select the pair which has the relation r* with highest score compared to all other pairs. The probability distribution of this pair will be used as the probability distribution of all event mention pairs between the two events.

In both approaches, we assign the *overlap* relations to all pairs of event mentions in the same event with probability 1.0.

Data	#Intervals	#E-mentions	#E-T	#E-E
Initial	232	324	305	376
Saturated	232	324	324	5940

Table 5.3: The statistics of our experimental data set.

5.7 Experimental Study

We first describe the experimental data and then present and discuss the experimental results.

5.7.1 Data and Setup

Most previous works in temporal reasoning used the TimeBank corpus as a benchmark. The corpus contains a fairly diverse collection of annotated event mentions, without any specific focus on certain event types. According to the annotation guideline of the corpus, most of verbs, nominalizations, adjectives, predicative clauses and prepositional phrases can be tagged as events. However, in practice, when performing temporal reasoning about events in a given text, one is typically only interested in significant and typed events, such as *Killing*, *Legislation*, *Election*. Furthermore, event mentions in TimeBank are neither annotated with their event arguments nor with event coreference information.

We noticed that the ACE 2005 corpus contains the annotation that we are interested in. The corpus consists of articles annotated with event mentions (with event triggers and arguments) and event coreference information. To create an experimental data set for our work, we selected from the corpus 20 newswire articles published in March 2003. To extract time intervals from the articles, we used the time interval extractor described in Section 5.4 with minimal post-processing. Implicit intervals are also added according to Section 5.3.2. We then hired an annotator with expertise in the field to annotate the data with the following information: (i) event mention and time interval association, and (ii) the temporal relations between event mentions, including $\{\bar{b}, \bar{a}, \bar{o}\}$. The annotator was not required to annotate all pairs of event mentions, but as many as possible. Next, we saturated the relations based on the initial annotations as follows: (i) event mentions that had not been associated with any time intervals were assigned to the entire timeline interval $(-\infty, +\infty)$, and (ii) added inferred temporal relations between event mentions with reflectivity and transitivity. Table 5.3 shows the data statistics before and after saturation. We note that in a

	Model	C_{E-T}			C_{E-E}			**Overall**		
		Prec.	**Rec.**	F_1	**Prec.**	**Rec.**	F_1	**Prec.**	**Rec.**	F_1
1	**Baseline**	33.29	33.29	33.29	20.86	32.81	25.03	27.06	33.05	29.16
2	**No Event Coref.**									
	Local classifiers	62.70	34.50	43.29	40.46	42.42	40.96	51.58	38.46	42.13
	Global inference	47.88	47.88	47.88	41.42	48.04	44.14	44.65	47.96	46.01
3	**Gold Event Coref.**									
	Local classifiers	50.88	50.88	50.88	43.86	52.65	47.46	47.37	51.77	49.17
	Global inference	50.88	50.88	50.88	48.04	62.45	54.05	49.46	56.67	52.47
4	**Learned Event Coref.**									
	Local classifiers	46.37	46.37	46.37	40.83	45.28	42.60	43.60	45.83	44.49
	Global inference	46.37	46.37	46.37	42.09	52.50	46.47	44.23	49.44	46.42

Table 5.4: Performance under various evaluation settings. All figures are averaged scores from 5-fold cross-validation experiments.

separate experiment, we still evaluated C_{E-E} on the TimeBank corpus and got better performance than a corresponding classifier in an existing work (see Section 5.7.4).

We conducted all experiments with 5-fold cross validation on our data set after saturation. The results of the systems are reported in averaged precision, recall and F_1 score on the association performance, for C_{E-T}, and the temporal relations, excluded the $\bar{n}$ relation, for C_{E-E}. We also measured the overall performance of the systems by computing the average of the performance of the classifiers.

5.7.2 A Baseline

We developed a baseline system that works as follows. It associates an event mention to the closest time interval found in the same sentence. If such an interval is not found, the baseline associates the mention with the closest time interval to the left. If the interval is again not found, the mention will be associated with the DCT interval. For the temporal relation between a pair of event mentions, the baseline treats the event mention that appears earlier in the text as temporally happening before the other mention. The baseline performance is shown in the first group of results in Table 5.4.

5.7.3 Our Systems

For our systems, we first evaluated the performance of our local pairwise classifiers and the global inference model. The second group of results in Table 5.4 shows the systems' performance. Our

global inference model relatively outperformed the baseline and the local classifiers significantly by 57.8% and 9.2% in F_1, respectively. This result shows the contribution of the constraints in our joint model.

In the case that the association between event mentions and time intervals are given (i.e. gold $E-T$), the overall performance of the local classifiers achieved 70.48 of F_1 score. The global inference in this setting boosted the overall performance to 80.98 F_1. On the other hand, when we provided the labels of the temporal relations between event mentions and put most weight on C_{E-E}, the overall performances of the local classifiers and the global inference model are 71.65 and 77.92, respectively.

Next, we integrated event coreference knowledge into our systems (as described in Section 5.6) and evaluated their performance. We experimented and observed that the *SumScore* approach works better for C_{E-T}, while *MaxScore* is more suitable for C_{E-E}. Our observations showed that event mentions of an event may appear in close proximity with multiple time intervals in the text, making C_{E-T} produce high prediction scores for many event mention-interval pairs. This, consequently, confuses *MaxScore* on the best association of the event and the time intervals, whereas *SumScore* overcomes the problem by averaging out the association scores. On the other hand, C_{E-E} gets more benefit from *MaxScore* because C_{E-E} works better on pairs of event mentions that appear closely in the text, which activate more valuable learning features. We will report the results using the best approach of each classifier.

To evaluate our systems with event coreference knowledge, we first experimented our systems with gold event coreference as given by the ACE 2005 corpus. Table 5.4 shows the contribution of event coreference to our systems in the third group of the results. The results show that injecting knowledge from event coreference remarkably improved both the local classifiers and the joint inference model. Overall, the system that combined event coreference and the global inference model achieved the best performance, which significantly overtook all other compared systems. Specifically, it outperformed the baseline system, the local classifiers, and the joint inference model without event coreference with 80%, 25%, and 14% of relative improvement in F_1, respectively. It also consistently outperformed the local classifiers enhanced with event coreference. We note that the precision and recall of C_{E-T} in the joint inference model are the same because the inference

model enforced each event mention to be associated with exactly one time interval. This is also true for the systems integrated with event coreference because our integration approaches assign only one time interval to an event mention.

We next move to experimenting with automatically learned event coreference systems. In this experiment, we re-trained the event coreference system described in [Chen et al., 2009] on all articles in the ACE 2005 corpus subtracting the 20 articles used in our data set. The performance of these systems are shown in the fourth group of the results in Table 5.4. The results show that by using a learned event coreference system, we achieved the same improvement trends as with gold event coreference. However, we did not obtain significant improvement when comparing with global inference without event coreference information. This result shows that the performance of an event coreference system can have a significant impact on the overall performance. While this suggests that a better event coreference system could potentially help the task more, it also opens the question whether event coreference can be benefited from our local classifiers through the use of a joint inference framework. We would like to leave this for future investigations.

5.7.4 Previous Work-Related Experiments

We also performed experiments using the same setting as in [Yoshikawa et al., 2009], which followed the guidelines of the TempEval challenges [Verhagen et al., 2007, Verhagen et al., 2010], on our saturated data. In these settings, we only identify temporal relations between events and temporal expressions that occur within the same sentence. In addition, only events of adjacent sentences are considered to perform temporal relation classification. We performed 5-fold cross validation without event coreference. Overall, the system achieved 29.99 F_1 for the local classifiers and 34.69 when the global inference is used. These results are better than the baseline but underperform our full models where those simplification assumptions are not imposed, as shown in Table 5.4, indicating the importance of relaxing their assumptions in practice.

We also evaluated our C_{E-E} on the TimeBank corpus. To prepare the evaluation data for this experiment, we followed the settings of [Chambers and Jurafsky, 2008a] to extract all event mention pairs that were annotated with *before* (or *ibefore*, "immediately before") and *after* (or *iafter*, "immediately after") relations in 183 news articles in the corpus. We trained and evaluated

our C_{E-E} on these examples with the same feature set that we evaluated in our experiments above, with gold tense and aspect features but without event type. Following their work, we performed 10-fold cross validation. Our classifier achieved a micro-averaged accuracy of 73.45%, whereas [Chambers and Jurafsky, 2008a] reported 66.8%. We next injected the knowledge of an event coreference system trained on the ACE2005 corpus into our C_{E-E}, and obtained a micro-averaged accuracy of 73.39%. It was not surprising that event coreference did not help in this dataset because: (i) different domains – the coreference was trained on ACE 05 but applied on TimeBank, and (ii) different annotation guidelines on events in ACE 2005 and TimeBank.

Finally, we conducted an experiment that justifies the advantages of our interval-based inference model over a time point-based inference. To do this, we first converted our data in Table 5.3 from intervals to time points and infer the temporal relations between the annotated event mentions and the time points: *before*, *after*, *overlap*, and *unknown*. We modified the first component in the objective function in (5.3) to accommodate these temporal relations. We also made several changes to the constraints, including removing those in (5.7) since they are no longer required, and adding constraints that ensure the relation between a time point and an event mention takes exactly one value. Proper changes were also made to other constraints in (5.10) to reflect the fact that time points are considered rather than intervals. We observed that experiment with such a formulation was unable to finish within 5 hours (we terminated the ILP inference after waiting for 5 hours), whereas our interval-based model finished the experiment with an average of 21 seconds per article.

5.8 Summary

We proposed an interval-based representation of the timeline of event mentions in an article. Our representation allowed us to formalize the joint inference model that can be solved efficiently, compared to a time point-based inference model, thus opening up the possibility of building more practical event temporal inference systems. Our inference model achieved significant improvement over the local classifiers. We also showed that event coreference can naturally support timeline construction, and good event coreference led to significant improvement in the system performance. Specifically, when such gold event coreference knowledge was injected into the model, a significant improvement in the overall performance could be obtained. While our experiments suggest that

the temporal classifiers can potentially help enhance the performance of event coreference, in future work we would like to investigate in coupling event coreference with other components in a global inference framework.

Chapter 6

Conclusions

The rapid development of advanced research in Natural Language Processing and Machine Learning has supported researchers to propose effective solutions for information extraction tasks. Specifically, this research direction concerns about algorithms that allow recognizing specific items of interest in unstructured text and discovering relations between them in order to extract and construct structured knowledge bases. Especially, with the emergence of large-scale IE systems, such as works in Open IE [Banko et al., 2007] and never-ending language learning framework [Carlson et al., 2010], the role of information extraction in general (and relation extraction in particular) becomes more and more important. It is well-known that previous work in this field has focused primarily on extracting information/relations of interest by using the evidences written explicitly in a given text. Unfortunately, despite the fact that this approach may achieve reasonable performances [Jiang, 2012], it is still very likely to be suffered from the problems of poor information representation and lack of reasoning knowledge.

In this thesis, we argued that relation extraction systems need to consult background knowledge, which is not expressed explicitly in an input text, in order to achieve a higher level of performance. We showed that background knowledge contributes positive impact on extraction systems. Moreover, we observed that using a single knowledge source may be not sufficient for an extraction system to achieve state-of-the-art performance. We, therefore, combined multiple knowledge sources into a system to achieve further gains. To this end, we proposed a principled framework that combines and integrates multiple sources of background knowledge into relation extraction. Our framework makes use of background knowledge in two ways:

- *Enriching inputs:* In this approach, background knowledge is used to map tasks' inputs to informative representations before being passed to core extraction algorithms.

- *Biasing outputs:* The outputs of an extraction systems are examined with background knowledge and enforced to follow global consistency criteria which may be conveyed through relational constraints in joint inference models.

It is possible to combine these two ways into a pipeline of a whole extraction system. Specifically, the inputs of a task is first mapped to an informative input space with background knowledge. After that, the core extraction algorithm will carry out the extraction process. Finally, the outputs from the core algorithm are corrected using background knowledge in an inference model. In this thesis, we studied the use of background knowledge and demonstrated its contributions in the context of several important learning-based relation extraction tasks as follows:

- **Taxonomic Relation Identification:** Identifying the taxonomic relation between any two input terms. We covered four basic relations: *ancestor*, *descendant*, *sibling* and *no relation*. We argued that without using a background knowledge resource to represent the input terms, they are less informative and, therefore, their relation could not be recognized effectively. Intuitively, we proposed algorithmic approaches that mapped input terms into another representation space supported by an external knowledge source so that we can extract meaningful features for the classification problem. Furthermore, we studied relational constraints as an additional knowledge source that enforced global coherence among local relation decisions to further improve the system performance. See Chapter 3.

- **Event Relation Discovery:** In chapter 4, we studied the problem of recognizing causality relation between events in news articles. In this task, we proposed not only a new metric to measure causality association strength between events but also a joint inference model that allowed us to incorporate knowledge from an additional knowledge source to achieve better experimental results.

- **Event Timeline Construction:** Our work in on this task mainly concerned about discovering the temporal order among events in news text. We constructed local classifiers to perform temporal reasoning on events in order to map them into an absolute timeline. Interestingly, we showed that by incorporating knowledge provided by event coreference we can achieve a

remarkable improvement over the performance of the local classifiers and the joint inference model that coupled the classifiers as well. See Chapter 5.

In summary, we showed that background knowledge sources play important roles in supporting and boosting relation extraction systems to achieving higher performance levels. Our framework allows one to combine multiple background knowledge sources that together contributing significantly positive impact on extraction systems. Nevertheless, we believe that we still can explore more benefits from background knowledge. Out experiments showed that there is still a big gap from what a system, which was enhanced by background knowledge, can achieve to the human performance. To make the gap become smaller, one may want to look at several other angles of a task and make use of background knowledge correspondingly to address each small phenomenon separately before combining them together in a larger joint inference model. Furthermore, it is equally important to address the question of how to leverage background knowledge sources effectively.

Appendix A

Semantic Classes

In this appendix, we list all the semantic classes that are used in **Dataset-I** and **Dataset-II** on the work of Taxonomic Relation Identification (see Chapter 3).

Below are the 40 semantic classes of **Dataset-I**:

actor, aircraftmodel, award, basicfood, carmodel, cartooncharacter, cellphonemodel, chemicalelem, city, company, country, currency, digitalcamera, disease, drug, empire, flower, holiday, hurricane, mountain, movie, nationalpark, nbateam, newspaper, painter, proglanguage, religion, river, searchengine, skybody, skyscraper, soccerclub, sportevent, stadium, terroristgroup, treaty, university, videogame, wine, worldwarbattle,

Here are the 44 semantic classes of **Dataset-II**:

academy award, airline, archbishop, artist, astronomer, author, boxing, city, cocktail, company, composer, country, county, department, discipline, driver, element, emperor, first lady, football, foundation, general, god, hockey, island, martial art, mountain range, pianist, poet, prefecture, president, prime minister, province, region, river, scientist, songwriter, stadium, star, superhero, tennis player, theatre, university, water,

Appendix B

Taxonomic Relational Constraints

In this appendix, we present the list of 35 manually-constructed relational constraints that were used in the global inference model in our work on taxonomic relation identification (see Chapter 3).

Each constraint is defined as an unlexicalized illegitimate term network that consists of 2 input terms x and y, and an additional term z. Constraints are written in a clockwise direction, starting from x and y. For instance, the illegitimate structure in Figure (c) forms the following relational constraint: $\langle \leftrightarrow, \leftrightarrow, \rightarrow \rangle$, where the arrows follow the notation in Table 3.1.

$\langle \leftarrow, \leftrightarrow, \not\leftrightarrow \rangle$, $\langle \leftarrow, \not\leftrightarrow, \leftrightarrow \rangle$, $\langle \leftrightarrow, \leftarrow, \leftarrow \rangle$, $\langle \leftrightarrow, \leftrightarrow, \leftarrow \rangle$, $\langle \rightarrow, \leftrightarrow, \not\leftrightarrow \rangle$, $\langle \leftrightarrow, \leftrightarrow, \not\leftrightarrow \rangle$, $\langle \not\leftrightarrow, \rightarrow, \leftarrow \rangle$, $\langle \leftarrow, \leftrightarrow, \leftarrow \rangle$, $\langle \leftrightarrow, \not\leftrightarrow, \not\leftrightarrow \rangle$, $\langle \not\leftrightarrow, \leftrightarrow, \leftrightarrow \rangle$, $\langle \not\leftrightarrow, \leftrightarrow, \rightarrow \rangle$, $\langle \not\leftrightarrow, \rightarrow, \leftrightarrow \rangle$, $\langle \not\leftrightarrow, \leftrightarrow, \leftarrow \rangle$, $\langle \not\leftrightarrow, \not\leftrightarrow, \not\leftrightarrow \rangle$, $\langle \rightarrow, \not\leftrightarrow, \leftarrow \rangle$, $\langle \leftrightarrow, \not\leftrightarrow, \leftrightarrow \rangle$, $\langle \leftrightarrow, \leftarrow, \rightarrow \rangle$, $\langle \leftrightarrow, \leftarrow, \leftrightarrow \rangle$, $\langle \not\leftrightarrow, \leftarrow, \leftrightarrow \rangle$, $\langle \not\leftrightarrow, \leftarrow, \rightarrow \rangle$, $\langle \rightarrow, \leftarrow, \leftrightarrow \rangle$, $\langle \rightarrow, \leftrightarrow, \leftrightarrow \rangle$, $\langle \leftrightarrow, \rightarrow, \leftrightarrow \rangle$, $\langle \leftarrow, \rightarrow, \leftrightarrow \rangle$, $\langle \leftarrow, \leftrightarrow, \leftrightarrow \rangle$, $\langle \rightarrow, \leftarrow, \not\leftrightarrow \rangle$, $\langle \leftarrow, \not\leftrightarrow, \leftarrow \rangle$, $\langle \leftarrow, \rightarrow, \not\leftrightarrow \rangle$, $\langle \rightarrow, \leftrightarrow, \rightarrow \rangle$, $\langle \leftrightarrow, \rightarrow, \not\leftrightarrow \rangle$, $\langle \leftrightarrow, \rightarrow, \leftarrow \rangle$, $\langle \leftarrow, \rightarrow, \leftarrow \rangle$, $\langle \leftrightarrow, \leftrightarrow, \rightarrow \rangle$, $\langle \not\leftrightarrow, \rightarrow, \rightarrow \rangle$, $\langle \leftarrow, \leftarrow, \leftarrow \rangle$

References

[Abad et al., 2010] Abad, A., Bentivogli, L., Dagan, I., Giampiccolo, D., Mirkin, S., Pianta, E., and Stern, A. (2010). A resource for investigating the impact of anaphora and coreference on inference. In *LREC*.

[Ahn, 2006] Ahn, D. (2006). The stages of event extraction. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*.

[Allen, 1983] Allen, J. F. (1983). Maintaining knowledge about temporal intervals. *Commun. ACM*.

[Alonso et al., 2011] Alonso, O., Strötgen, J., Baeza-Yates, R., and Gertz, M. (2011). Temporal information retrieval: Challenges and opportunities. In *TWAW*.

[Amigo et al., 2011] Amigo, E., Artiles, J., Li, Q., and Ji, H. (2011). An evaluation framework for aggregated temporal information extraction. In *Proc. SIGIR Workshop on Entity-Oriented Search*.

[Banko et al., 2007] Banko, M., Cafarella, M., Soderland, M., Broadhead, M., and Etzioni, O. (2007). Open information extraction from the web. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2670–2676.

[Banko and Etzioni, 2008] Banko, M. and Etzioni, O. (2008). The tradeoffs between open and traditional relation extraction. In *ACL-HLT*.

[Baroni and Lenci, 2010] Baroni, M. and Lenci, A. (2010). Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36.

[Beamer and Girju, 2009] Beamer, B. and Girju, R. (2009). Using a bigram event model to predict causal potential. In *Proceedings of CICLING-09*.

[Bethard and Martin, 2006] Bethard, S. and Martin, J. H. (2006). Identification of event mentions and their semantic class. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*.

[Bethard et al., 2007a] Bethard, S., Martin, J. H., and Klingenstein, S. (2007a). Timelines from text: Identification of syntactic temporal relations. In *Proceedings of the International Conference on Semantic Computing*.

[Bethard et al., 2007b] Bethard, S., Martin, J. H., and Klingenstein, S. (2007b). Timelines from text: Identification of syntactic temporal relations. *International Conference on Semantic Computing*.

[Bishop, 1996] Bishop, C. M. (1996). *Neural networks for pattern recognition*. Oxford University Press, Oxford, UK.

[Bramsen et al., 2006] Bramsen, P., Deshpande, P., Lee, Y. K., and Barzilay, R. (2006). Inducing temporal graphs. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, pages 189–198, Sydney, Australia. Association for Computational Linguistics.

[Bunescu and Mooney, 2006] Bunescu, R. and Mooney, R. (2006). Subsequence kernels for relation extraction. In Weiss, Y., Schölkopf, B., and Platt, J., editors, *Advances in Neural Information Processing Systems 18*, pages 171–178. MIT Press, Cambridge, MA.

[Bunescu and Mooney, 2005] Bunescu, R. C. and Mooney, R. J. (2005). A shortest path dependency kernel for relation extraction. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 724–731, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Carlson et al., 2010] Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Jr., E. R. H., and Mitchell, T. M. (2010). Toward an architecture for never-ending language learning. In *Proceedings of the Twenty-Fourth Conference on Artificial Intelligence (AAAI 2010)*.

[Chakrabarti et al., 1997] Chakrabarti, S., Dom, B., Agrawal, R., and Raghavan, P. (1997). Using taxonomy, discriminants, and signatures for navigating in text databases. In *VLDB*.

[Chambers and Jurafsky, 2008a] Chambers, N. and Jurafsky, D. (2008a). Jointly combining implicit constraints improves temporal ordering. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*.

[Chambers and Jurafsky, 2008b] Chambers, N. and Jurafsky, D. (2008b). Unsupervised learning of narrative event chains. In *ACL-HLT*.

[Chambers and Jurafsky, 2009] Chambers, N. and Jurafsky, D. (2009). Unsupervised learning of narrative schemas and their participants. In *ACL*.

[Chan and Roth, 2010] Chan, Y. and Roth, D. (2010). Exploiting background knowledge for relation extraction. In *Proceedings the International Conference on Computational Linguistics (COLING)*, Beijing, China.

[Chang et al., 2008a] Chang, M., Ratinov, L., Rizzolo, N., and Roth, D. (2008a). Learning and inference with constraints. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 1513–1518.

[Chang et al., 2007] Chang, M., Ratinov, L., and Roth, D. (2007). Guiding semi-supervision with constraint-driven learning. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 280–287, Prague, Czech Republic. Association for Computational Linguistics.

[Chang et al., 2008b] Chang, M., Ratinov, L., and Roth, D. (2008b). Constraints as prior knowledge. In *ICML Workshop on Prior Knowledge for Text and Language Processing*, pages 32–39.

[Charniak and Johnson, 2005] Charniak, E. and Johnson, M. (2005). Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 173–180, Ann Arbor, Michigan. ACL.

[Chen et al., 2009] Chen, Z., Ji, H., and Haralick, R. (2009). A pairwise event coreference model, feature impact and evaluation for event coreference resolution. In *Proceedings of the Workshop on Events in Emerging Text Types*. Association for Computational Linguistics.

[Clarke and Lapata, 2008] Clarke, J. and Lapata, M. (2008). Global inference for sentence compression: An integer linear programming approach. *Journal of Artificial Intelligence Research (JAIR)*, 31:399–429.

[Culotta and Sorensen, 2004] Culotta, A. and Sorensen, J. (2004). Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Dagan et al., 2006] Dagan, I., Glickman, O., and Magnini, B., editors (2006). *The PASCAL Recognising Textual Entailment Challenge.*, volume 3944. Springer-Verlag, Berlin.

[Davidov and Rappoport, 2008a] Davidov, D. and Rappoport, A. (2008a). Unsupervised discovery of generic relationships using pattern clusters and its evaluation by automatically generated sat analogy questions. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

[Davidov and Rappoport, 2008b] Davidov, D. and Rappoport, A. (2008b). Unsupervised discovery of generic relationships using pattern clusters and its evaluation by automatically generated SAT analogy questions. In *ACL-HLT*.

[Denis and Baldridge, 2007] Denis, P. and Baldridge, J. (2007). Joint determination of anaphoricity and coreference resolution using integer programming. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*, pages 236–243, Rochester, New York. Association for Computational Linguistics.

[Denis and Muller, 2011a] Denis, P. and Muller, P. (2011a). Predicting globally-coherent temporal structures from texts via endpoint inference and graph decomposition. In *IJCAI*.

[Denis and Muller, 2011b] Denis, P. and Muller, P. (2011b). Predicting globally-coherent temporal structures from texts via endpoint inference and graph decomposition. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*.

[Do et al., 2011] Do, Q., Chan, Y. S., and Roth, D. (2011). Minimally supervised event causality extraction. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, Edinburgh, Scotland.

[Do and Roth, 2010] Do, Q. and Roth, D. (2010). Constraints based taxonomic relation classification. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, pages 1099–1109, Massachusetts, USA.

[Do and Roth, 2012a] Do, Q. and Roth, D. (2012a). Exploiting the wikipedia structure in local and global classification of taxonomic relations. *Journal of Natural Language Engineering, Special Issue on Statistical Learning of Natural Language Structured Input and Output*, 18.

[Do and Roth, 2012b] Do, Q. and Roth, D. (2012b). Joint inference for event timeline construction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and Conference on Computational Natural Language Learning (EMNLP-CoNLL). In submission.*

[Fader et al., 2011] Fader, A., Soderland, S., and Etzioni, O. (2011). Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*.

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

[Freund and Schapire, 1999] Freund, Y. and Schapire, R. E. (1999). Large margin classification using the perceptron algorithm. *Machine Learning*, 37(3):277–296.

[Gabrilovich and Markovitch, 2007] Gabrilovich, E. and Markovitch, S. (2007). Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1606–1611.

[Girju, 2003] Girju, R. (2003). Automatic detection of causal relations for question answering. In *ACL workshop on Multilingual Summarization and Question Answering*.

[Grishman et al., 2002] Grishman, R., Huttunen, S., and Yangarber, R. (2002). Real-time event extraction for infectious disease outbreaks. In *Proceedings of the second international conference on Human Language Technology Research (HLT)*.

[Grishman et al., 2005] Grishman, R., Westbrook, D., and Meyers, A. (2005). NYU's English ACE 2005 system description. Technical report, Department of Computer Science, New York University.

[Gurevich et al., 2008] Gurevich, O., Crouch, R., King, T. H., and de Paiva, V. (2008). Deverbal nouns in knowledge representation. *Journal of Logic and Computation*, 18.

[Hardy et al., 2006] Hardy, H., Kanchakouskaya, V., and Strzalkowski, T. (2006). Automatic event classification using surface text features. In *Proceedings of the 2006 AAAI Workshop on EventExtractionand Synthesis*.

[Hearst, 1992a] Hearst, M. A. (1992a). Acquisition of hyponyms from large text corpora. In *COLING*.

[Hearst, 1992b] Hearst, M. A. (1992b). Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics*, pages 539–545, Morristown, NJ, USA. Association for Computational Linguistics.

[Hotho et al., 2003] Hotho, A., Staab, S., and Stumme, G. (2003). Ontologies improve text document clustering. In *ICDM*.

[Ji and Grishman, 2008] Ji, H. and Grishman, R. (2008). Refining event extraction through cross-document inference. In *Proceedings of ACL-08: HLT*.

[Ji et al., 2011] Ji, H., Grishman, R., and Dang, H. T. (2011). Overview of the tac2011 knowledge base population track. In *TAC*.

[Jiang, 2012] Jiang, J. (2012). Information extraction from text. In *Mining Text Data*, pages 11–41. Springer.

[Jiang and Zhai, 2007] Jiang, J. and Zhai, C. (2007). A systematic exploration of the feature space for relation extraction. In *Proceeding of NAACL-HLT*, pages 113–120, Rochester, New York. Association for Computational Linguistics.

[Kozareva et al., 2008] Kozareva, Z., Riloff, E., and Hovy, E. (2008). Semantic class learning from the web with hyponym pattern linkage graphs. In *ACL-HLT*.

[Leacock and Chodorow, 1998] Leacock, C. and Chodorow, M. (1998). *Combining Local Context and WordNet Similarity for Word Sense Identification*, chapter 11, pages 265–284. MIT Press.

[Li et al., 2011] Li, H., Ji, H., Deng, H., and Han, J. (2011). Exploiting background information networks to enhance bilingual event extraction through topic modeling. In *Proc. of International Conference on Advances in Information Mining and Management (IMMM)*.

[Lin et al., 2010] Lin, Z., Ng, H. T., and Kan, M.-Y. (2010). A pdtb-styled end-to-end discourse parser. Technical report, National University of Singapore. http://www.comp.nus.edu.sg/ linzihen/publications/tech2010.pdf.

[MacCartney and Manning, 2008] MacCartney, B. and Manning, C. D. (2008). Modeling semantic containment and exclusion in natural language inference. In *COLING*.

[Mani et al., 2006a] Mani, I., Verhagen, M., Wellner, B., Lee, C. M., and Pustejovsky, J. (2006a). Machine learning of temporal relations. In *ACL*.

[Mani et al., 2006b] Mani, I., Verhagen, M., Wellner, B., Lee, C. M., and Pustejovsky, J. (2006b). Machine learning of temporal relations. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL)*.

[Marneffe et al., 2006] Marneffe, M. D., Maccartney, B., and Manning, C. D. (2006). Generating typed dependency parses from phrase structure parses. In *In LREC 2006*.

[Martins et al., 2009] Martins, A., Smith, N. A., and Xing, E. (2009). Concise integer linear programming formulations for dependency parsing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

[Mihalcea and Csomai, 2007] Mihalcea, R. and Csomai, A. (2007). Wikify!: linking documents to encyclopedic knowledge. In *CIKM*.

[Milne and Witten, 2008] Milne, D. and Witten, I. H. (2008). Learning to link with wikipedia. In *CIKM*.

[Munoz et al., 1999] Munoz, M., Punyakanok, V., Roth, D., and Zimak, D. (1999). A learning approach to shallow parsing. Technical Report UIUCDCS-R-99-2087, UIUC Computer Science Department.

[Paşca, 2007] Paşca, M. (2007). Organizing and searching the world wide web of facts step two: Harnessing the wisdom of the crowds. In *WWW*.

[Paşca and Van Durme, 2008] Paşca, M. and Van Durme, B. (2008). Weakly-supervised acquisition of open-domain classes and class attributes from web documents and query logs. In *ACL-HLT*.

[Padó and Lapata, 2007] Padó, S. and Lapata, M. (2007). Dependency-based construction of semantic space models. *Computational Linguistics*.

[Pantel and Pennacchiotti, 2006] Pantel, P. and Pennacchiotti, M. (2006). Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 113–120.

[Ponzetto and Strube, 2007] Ponzetto, S. P. and Strube, M. (2007). Deriving a large scale taxonomy from wikipedia. *AAAI*.

[Prasad et al., 2007] Prasad, R., Miltsakaki, E., Dinesh, N., Lee, A., Joshi, A., Robaldo, L., and Webber, B. (2007). The penn discourse treebank 2.0 annotation manual. Technical report, University of Pennsylvania.

[Punyakanok et al., 2008] Punyakanok, V., Roth, D., and Yih, W. (2008). The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, 34(2):257–287.

[Pustejovsky et al., 2003a] Pustejovsky, J., Castano, J., Ingria, R., Sauri, R., Gaizauskas, R., Setzer, A., and Katz, G. (2003a). Timeml: Robust specification of event and temporal expressions in text. In *IWCS-5*.

[Pustejovsky et al., 2003b] Pustejovsky, J., Hanks, P., Sauri, R., See, A., Gaizauskas, R., Setzer, A., Radev, D., Sundheim, B., Day, D., Ferro, L., and Lazo, M. (2003b). The TIMEBANK corpus. In *Proceedings of Corpus Linguistics 2003*.

[Ramshaw and Marcus, 1995] Ramshaw, L. A. and Marcus, M. P. (1995). Text chunking using transformation-based learning. In *Proceedings of the Third Annual Workshop on Very Large Corpora*.

[Ratinov et al., 2011] Ratinov, L., Downey, D., Anderson, M., and Roth, D. (2011). Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

[Riaz and Girju, 2010] Riaz, M. and Girju, R. (2010). Another look at causality: Discovering scenario-specific contingency relationships with no supervision. *International Conference on Semantic Computing*.

[Ritter et al., 2008] Ritter, A., Soderland, S., Downey, D., and Etzioni, O. (2008). It's a contradiction – no, it's not: A case study using functional relations. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*.

[Rizzolo and Roth, 2010] Rizzolo, N. and Roth, D. (2010). Learning Based Java for Rapid Development of NLP Systems. In *Proceedings of the International Conference on Language Resources and Evaluation*, Valletta, Malta.

[Roth and Yih, 2004] Roth, D. and Yih, W. (2004). A linear programming formulation for global inference in natural language tasks. In Ng, H. T. and Riloff, E., editors, *Proceedings of the Annual Conference on Computational Natural Language Learning (CoNLL)*, pages 1–8. Association for Computational Linguistics.

[Roth and Yih, 2007] Roth, D. and Yih, W. (2007). Global inference for entity and relation identification via a linear programming formulation. In Getoor, L. and Taskar, B., editors, *Introduction to Statistical Relational Learning*. MIT Press.

[Roth and Zelenko, 1998] Roth, D. and Zelenko, D. (1998). Part of speech tagging using a network of linear separators. In *COLING-ACL, The 17th International Conference on Computational Linguistics*, pages 1136–1142.

[Ruppenhofer et al., 2010] Ruppenhofer, J., Ellsworth, M., Petruck, M. R. L., Johnson, C. R., and Scheffczyk, J. (2010). *FrameNet II: Extended Theory and Practice*. Unpublished.

[Sammons et al., 2010] Sammons, M., Vydiswaran, V. G. V., and Roth, D. (2010). "ask not what textual entailment can do for you...". In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1199–1208, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Sarmento et al., 2007] Sarmento, L., Jijkuon, V., de Rijke, M., and Oliveira, E. (2007). "more like these": growing entity classes from seeds. In *CIKM*.

[Saxena et al., 2007] Saxena, A. K., Sambhu, G. V., Kaushik, S., and Subramaniam, L. V. (2007). Iitd-ibmirl system for question answering using pattern matching, semantic type and semantic category recognition. In *TREC*.

[Sekine, 2006] Sekine, S. (2006). On-demand information extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 731–738.

[Shahaf and Guestrin, 2010] Shahaf, D. and Guestrin, C. (2010). Connecting the dots between news articles. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*.

[Snow et al., 2005] Snow, R., Jurafsky, D., and Ng, A. Y. (2005). Learning syntactic patterns for automatic hypernym discovery. In *NIPS*.

[Snow et al., 2006] Snow, R., Jurafsky, D., and Ng, A. Y. (2006). Semantic taxonomy induction from heterogenous evidence. In *ACL*.

[Strötgen and Gertz, 2010a] Strötgen, J. and Gertz, M. (2010a). Heideltime: High quality rule-based extraction and normalization of temporal expressions. In *Proceedings of the 5th International Workshop on Semantic Evaluation*.

[Strötgen and Gertz, 2010b] Strötgen, J. and Gertz, M. (2010b). Heideltime: High quality rule-based extraction and normalization of temporal expressions. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval '10.

[Suchanek et al., 2007] Suchanek, F. M., Kasneci, G., and Weikum, G. (2007). Yago: A Core of Semantic Knowledge. In *WWW*.

[Sun et al., 2007] Sun, Y., Liu, N., Xie, K., Yan, S., Zhang, B., and Chen, Z. (2007). Causal relation of queries from temporal logs. In *Proceedings of WWW-07*.

[Suppes, 1970] Suppes, P. (1970). *A Probabilistic Theory of Causality*. Amsterdam: North-Holland Publishing Company.

[Tanev et al., 2008] Tanev, H., Piskorski, J., and Atkinson, M. (2008). Real-time news event extraction for global crisis monitoring. In *Proceedings of the 13th international conference on Natural Language and Information Systems: Applications of Natural Language to Information Systems*.

[Tatu and Srikanth, 2008] Tatu, M. and Srikanth, M. (2008). Experiments with reasoning for temporal relations between events. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*.

[Turney and Pantel, 2010] Turney, P. D. and Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *JAIR*.

[Verhagen et al., 2007] Verhagen, M., Gaizauskas, R., Schilder, F., Hepple, M., Katz, G., and Pustejovsky, J. (2007). Semeval-2007 task 15: Tempeval temporal relation identification. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*.

[Verhagen et al., 2010] Verhagen, M., Sauri, R., Caselli, T., and Pustejovsky, J. (2010). Semeval-2010 task 13: Tempeval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*.

[Vikas et al., 2008] Vikas, O., Meshram, A. K., Meena, G., and Gupta, A. (2008). Multiple document summarization using principal component analysis incorporating semantic vector space model. In *Computational Linguistics and Chinese Language Processing*.

[Vyas and Pantel, 2009] Vyas, V. and Pantel, P. (2009). Semi-automatic entity set refinement. In *NAACL-HLT*.

[Wang et al., 2010] Wang, Y., Zhu, M., Qu, L., Spaniol, M., and Weikum, G. (2010). Timely yago: harvesting, querying, and visualizing temporal knowledge from wikipedia. In *EDBT*.

[Yoshikawa et al., 2009] Yoshikawa, K., Riedel, S., Asahara, M., and Matsumoto, Y. (2009). Jointly identifying temporal relations with markov logic. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*.

[Zelenko et al., 2003] Zelenko, D., Aone, C., and Richardella, A. (2003). Kernel methods for relation extraction. *Journal of Machine Learning Research*.

[Zhang et al., 2006] Zhang, M., Zhang, J., and Su, J. (2006). Exploring syntactic features for relation extraction using a convolution tree kernel. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL '06, pages 288–295, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Zhao et al., 2012] Zhao, R., Do, Q., and Roth, D. (2012). A robust shallow temporal reasoning system. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Demo Session (NAACL-Demo)*.

[Zhou et al., 2005] Zhou, G., Su, J., Zhang, J., and Zhang, M. (2005). Exploring various knowledge in relation extraction. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 427–434, Ann Arbor, Michigan. Association for Computational Linguistics.