NAIST-IS-DD0961024

Doctoral Dissertation

Probabilistic Logic Approach to Event Structure Analysis

Katsumasa Yoshikawa

September 22, 2011

Department of Information Processing
Graduate School of Information Science
Nara Institute of Science and Technology

Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Katsumasa Yoshikawa

Thesis Committee:	Professor Yuji Matsumoto	(Supervisor)
	Professor Hiroyuki Seki	(Co-supervisor)
	Associate Professor Masashi Shimbo	(Co-supervisor)
	Assistant Professor Masayuki Asahara	(Co-supervisor)
	Research Scientist Sebastian Riedel	(University of Massachusetts)

Probabilistic Logic Approach to Event Structure Analysis*

Katsumasa Yoshikawa

Abstract

This thesis describes a probabilistic approach to event structure analysis, a task of extracting events, arguments and their relations. An *event* refers to a change of state that constitutes a story in a document. Events play important roles in natural language documents. This thesis attempts to analyze such narrative events by combining humans' linguistic knowledge with probabilistic information from annotated corpora.

We focus our effort on constructing new analyzers using one of the most popular probabilistic logic frameworks: Markov Logic. Markov Logic is a combination of first order logic and Markov Networks. First order logic can efficiently implement humans' linguistic knowledges. Markov Networks allow us to exploit various features acquired from large corpora.

The biggest advantage of Markov Logic is that it can treat multiple decision simultaneously. In real world data, relations depend on each other in various ways. Suppose we want to identify the relations between predicate and their arguments in a sentence. Identification often fails if we only consider a pair of predicate-argument individually because these predicate-argument relations are often syntactically and semantically dependent of each other. We attempt to deal with such dependencies by describing humans' linguistic knowledge as Markov Logic formulae.

Our target task is relation extraction about events that behave as verb. In particular, we tackle the following three relation extraction tasks.

First, we perform predicate-argument relation extraction on Japanese newswire corpus. In this task, our method considers all words in a sentence and find the most possible assignments of predicate-argument. We also make qualitative analysis and confirm the effectiveness of our method.

*Doctoral Dissertation, Department of Information Processing, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD0961024, September 22, 2011.

Our second task is that event-argument relation extraction on biomedical corpus. We introduce coreference relations to extract event-argument relations. Transition rules with coreference relations allow us to extract event-argument relations crossing over sentence boundaries.

Third, we tackle temporal relation identification in which we identify temporal orders of events, time expressions, and document creation time. We implement logical constraints of temporal relations in Markov Logic and our model finds the optimal solutions in a document.

Our contributions are not limited to the performance improvements in particular tasks. Events and arguments which we focus on are important elements in a documents. Therefore, our work makes a nice foothold towards automatic understanding of documents.

Keywords:

markovl logic, semantic role labeling, event extraction, temporal relation identification, coreference resolution, transition rule

確率的論理による事象構造解析*

吉川 克正

内容梗概

本稿では確率的論理を利用した事象構造の解析を行い，事象と項及びその関係の抽出について述べる．事象とは，文書中における物語を構成する“状態変化”を表す表現である．この事象表現が自然言語の文書において，重要な役割を果たすことはよく知られている．本研究では，このような事象表現に対する解析を，人間の持つ言語学的知識とコーパスからの統計情報の組み合わせにより行うことを目的とする．

本研究で扱う確率的論理は，マルコフ論理と呼ばれ，広く利用されている枠組みである．マルコフ論理は一階述語論理とマルコフネットワークを組み合わせた枠組みであるため，一階述語論理により人間の持つ言語知識を効率的に実装するとともに，マルコフネットワークにより，コーパスから様々な素性を学習して利用することができる．本研究ではこのマルコフ論理によって自然言語処理における三つのタスクを扱い，人間の言語知識とコーパスからの学習を効果的に組み合わせるモデルを提案する．

マルコフ論理による最大の利点は，複数の決定を同時に行える点である．現実のデータでは，データ間に様々な依存関係が考えられる．例えば，一つの文内に複数の述語とそれに対応する項が存在する場合，ただ一つの述語-項の組み合わせだけに着目しても正しい解を得られないことが多くある．なぜなら，同一文内に存在する述語-項関係には互いに依存関係があるからである．本研究の目的は，このような依存関係を人間の知識に基づく論理式により捉えることで，様々な自然言語処理のタスクにおいてより効果的なモデルを構築することにある．

本稿で扱うタスクはいずれも事象に関する関係抽出である．事象の中でも，特に動詞として働くものを中心に考え，三つの関係抽出タスクを行っている．

まず一つ目は日本語の新聞記事における述語-項関係抽出である．文内全ての単語を同時に考慮して文全体で最適な述語-項関係を捉えるモデルを構築している．またこのタスクでは定量的な評価だけでなく，定性的な評価にも力を入れた．

*奈良先端科学技術大学院大学 情報科学研究科 情報処理学専攻 博士論文, NAIST-IS-DD0961024, 2011 年 9月 22 日.

二つ目は医学生物学文書における事象-項関係抽出である．このタスクにおける重要な点として，医学生物学文書という限定的な分野ではあるものの，共参照関係を事象-項関係抽出に利用することに成功したことが挙げられる．その結果，文境界を越えるような事象-項関係も扱うことができるようになっている．

三つ目は，時間順序関係推定と呼ばれ，事象表現，時間表現，文書作成日時に関してその時間的な順序を推定する問題である．時間的関係に対する大域的な論理制約を導入することで，文書全体における論理的整合性を向上させる試みである．

本研究の貢献は個々のタスクにおける性能の向上に止まらない．なぜなら本研究で扱う事象や項は，文書において主要な役割を果たす要素であるため，その関係を効果的に扱える方法を提案できたことは，文書理解において重要な足がかりを得たことに他ならないからである．

キーワード

マルコフロジック, 述語項構造解析, 事象抽出, 時間的順序関係推定, 共参照解析, 推移律

Acknowledgements

まず主指導教官である松本裕治教授には，修士からの 4 年間，大変お世話になりました．研究内容に対するご助言だけでなく，日頃の雑談の中で，言語処理研究の面白さ，そして研究者としての生き方を学ばせて頂きました．何より驚くほど制限のない，本当に自由な環境で研究をさせて頂けたことを，改めて深く感謝しています．共同研究などで，様々な方と仕事をしたいと考えていた私にとって，これ以上は望めない環境でした．ありがとうございます．

関浩之先生には，お忙しい中審査委員を引き受けて頂き，修士過程の頃から通して的確なコメントを頂きました．発表の度にかけて頂いた暖かいコメントが，研究を進める上で大きな励みとなりました．感謝いたします．

新保仁准教授には，研究の合間に笑い話をさせて頂く間柄でありながら，研究内容に関しては常に厳しいコメントを頂き，研究に臨む真摯な態度を教えて頂きました．目先を追うのに必死になっている時，ふと立ち止まって考え直すきっかけを下さったのはいつも新保さんでした．ありがとうございます．

浅原正幸助教には，何をしてよいのか分からない全くゼロの状態の自分に，時間情報解析という興味深いテーマを与えて頂き，特に修士過程の間には，密に研究の進め方を指導して頂きました．本当に感謝しています．自然言語処理という分野において，これからも研究を続けていきたいと思えるのは，間違いなく浅原さんのおかげです．今後ともよろしくお願いいたします．

I would like to express my sincere gratitude to Dr. Sebastian Riedel (University of Massachusetts, Amherst) for all his valuable advices and supports as a member of my thesis committee. I am also very happy about the joint work I have done with Seba-san. Many of the results in this thesis are thanks to interactions with him.

小町守助教には研究室の良き先輩として，研究における議論のみならず，インターンシップや就職活動などのご助言を多く頂きました．悩み多き博士過程の学生生活は，常に学生に近い目線で話をして下さった小町さんの助けなしに乗り切れませんでした．本当にありがとうございます．

東北大学の乾健太郎教授には，他ではできない言語の深い内容について，様々な議論を交わし，示唆に富んだコメントを多数頂きました．研究に対する溢れる

程の情熱と，それに全霊を傾けて取り組む乾さんのバイタリティにいつも刺激を受けていました．深く感謝いたします．

NTT コミュニケーション科学基礎研究所の平尾努さんには半年間のインターンシップでお世話になりました．その後も研究全般について，修了後の進路について，事あるごとに有用なコメントを頂きました．常に肩の力を抜いて柔軟に取り組むことの大切さを教えて頂いたこと，そして私の打たれ強さを誰よりも高く評価して下さったこと，ここに深く感謝いたします．

常日頃から研究についてよく議論させて頂くとともに，研究生活全般についても多く語り合う機会を頂いた東京工業大学の飯田龍助教，木村学研究員，東北大学の渡邉陽太郎助教に感謝いたします．毎週のようにカレーを作り，食べながら話したあの時間の中には，自分が大学院生として望んだ全てが詰まっていました．

日本酒を愛でる会を主催してくださった楽天技術研究所の村上浩司さんは，人生について学ばせて貰う良き兄貴であり，日々研究のプレッシャーと戦いながら，酒を酌み交わして癒し合う戦友でもありました．本当の意味で酒を楽しみ，うまくストレスと付き合う術は村上さんから学びました．感謝しています．

秘書の北川祐子さんには，研究活動における煩雑な事務手続きを一手に引き受けて頂き，感謝の念に堪えません．また研究の合間に聞かせて貰う世間話が，うまくいかない研究や就職活動の中，とてもよい息抜きとなりました．

研究室の同期として切磋琢磨した岩立将和さん，水野淳太さん，大熊秀治さん，清水友裕さん，森田啓さん，後輩であり，良き話し相手であった，林克彦さん，井之上直也さん，江口 萌さん，小嵜耕平さん，木曽鉄男さん，林部祐太さん，北裏龍太さん，後藤隼人さん，大木環美さん，その他，同じ研究室で過ごした多くの人達に感謝します．この研究室で，皆さんの研究意欲を肌で感じながら同じ時間を過ごせたこと，この上ない幸運であったと感じます．

最後に一度は就職した自分の大学院進学を応援し，暖かく見守り，支え続けてくれた母，末子と，妹，直美に深く感謝します．

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

Events and their arguments take important roles in natural language documents. Constructing effective analyzers for event structure is one of the significant tasks in Natural Language Processing (NLP). However, event structure analysis requires to take into account complex relations based on deep semantic knowledge. It it difficult for such tasks to apply automatic approach by machine learning. Machine learning approach has actually achieved good performance on a variety of tasks in NLP. In syntactic analysis, local information often find a correct answer. For Part-of-Speech (POS) tagging we can infer a correct tag of each token with only local information around the target tokens. However, some tasks require not only local but also global information to resolve. For example, in semantic role labeling that we need to predict a relation between verbs and their semantic arguments, some relations between verbs and their arguments depend on each other. In order to consider such dependencies, we need to make several decisions, jointly. But the predominant approaches to NLP are based on local classifiers and cannot jointly handle several decisions.

In addition, machine learning approach can collect a great number of features from large corpora and exploit them for decisions but a few pieces of humans' knowledges often overcome the vast features. Ideally, we would like to exploit both probabilistic models acquired from corpora and effective humans' knowledge. In recent years, probabilistic logic approach has come into active because it allows us to apply both merits of probabilistic models and humans' knowledges.

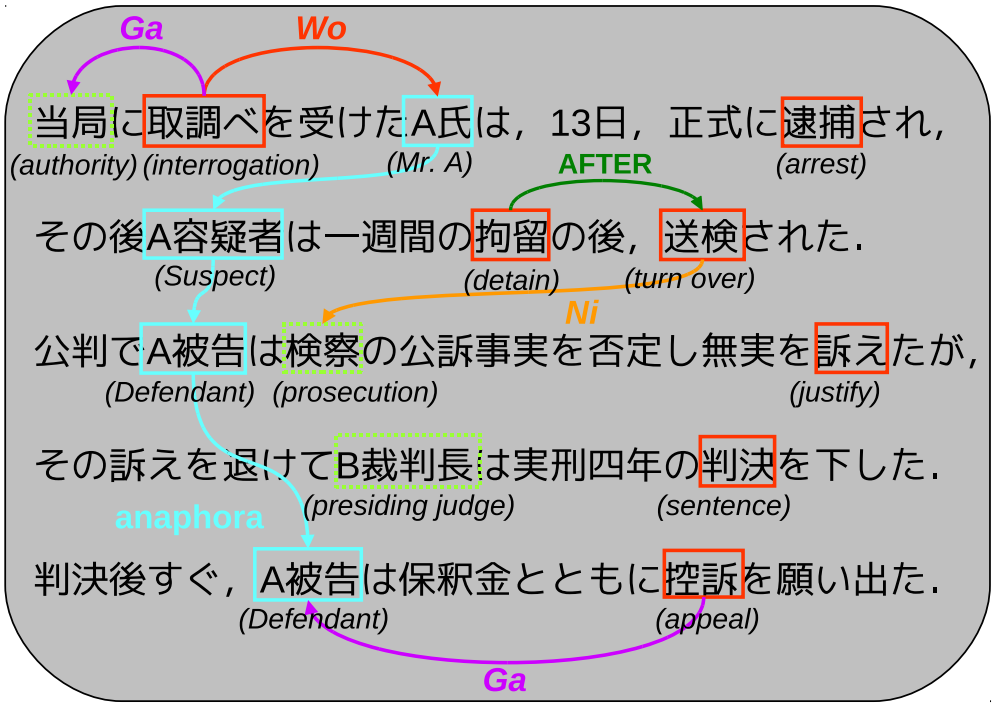


Figure 1.1: Events and Arguments in a Document

1.2 Research Goal

In this thesis, we describe probabilistic logic approach for event structure analysis. We focus our effort on relations about events and their arguments. Analysing events and their arguments contributes to understand document. Figure 1.1 shows an article which has events (e.g. '取調べ (interrogation)', '逮捕 (arrest)') and arguments (e.g. '当局 (authority)', 'A 氏 (Mr. A)'). All the relations we extracted from the article in Figure 1.1 are illustrated in Figure 1.2.

The relations in Figure 1.2 are divided into three types:

event-event temporal relation, causal relation (e.g., "拘留 (detain)" is AFTER "送検 (turn over)")

event-argument semantic role (e.g., "A 氏 (Mr. A)" is the Wo case (Accusative) of "取調べ (interrogation)")

argument-argument anaphora relation, coreference relation (e.g., "A 氏 (Mr. A)" is coreferent to "A 容疑者 (Suspect)").

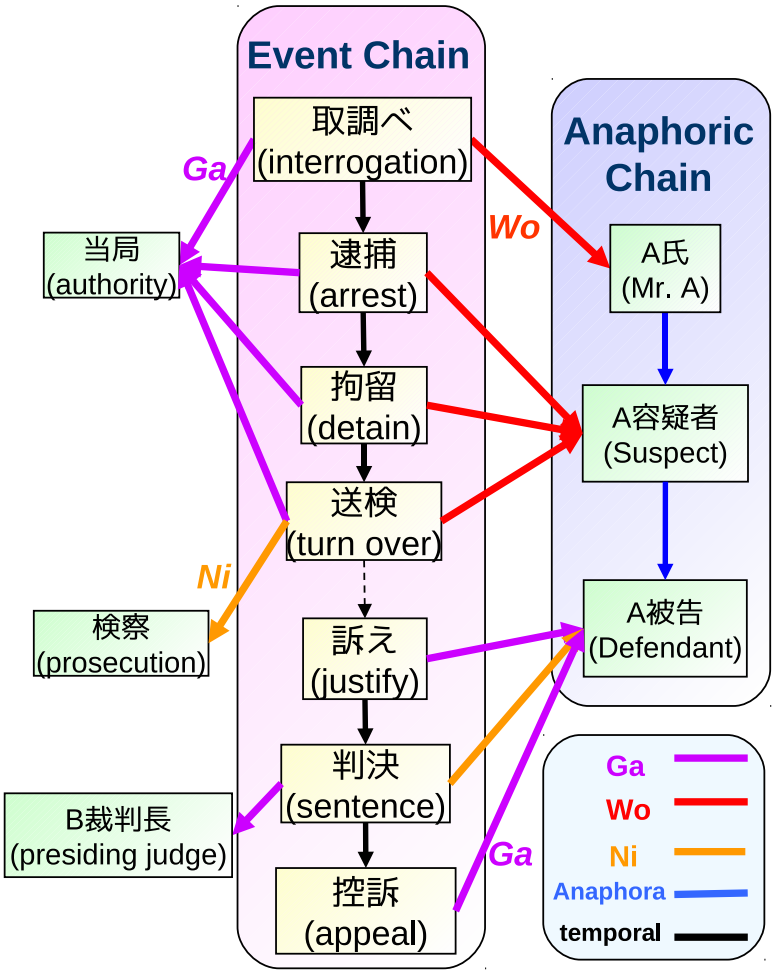


Figure 1.2: Events, arguments, and their relations extracted from Figure 1.1

In addition, we aim to propose new methods to extract these types of relations by logical constraints based on humans' linguistic knowledge. We apply probabilistic logic approaches for this purpose.

From another point of view, our objective is proposing new methods which give us a new foothold in document understanding. Needless to mention, an ultimate goal of NLP is automatic document understanding. As shown in Figure 1.2, events and arguments construct sequences called *event chain* and *anaphoric chain*. These chains depict the main story of the document and should contribute to understanding documents. As our future work we will exploit these chains for the applications such as document summarization, machine translation, and so on.

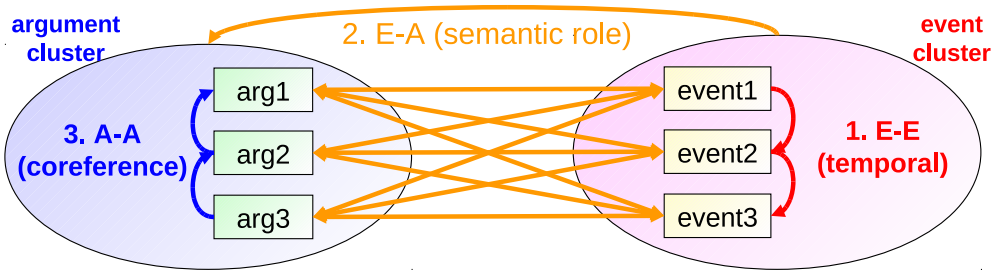


Figure 1.3: Target Relations

1.3 Probabilistic Logic

Probabilistic Logic is one of the Statistical Relational Learning frameworks which integrate probabilistic models and relational representations. In this thesis, we will exploit a particular probabilistic logic framework: Markov Logic (Richardson and Domingos, 2006) which combines Markov Network and First Order Logic. Markov Logic allows us to represent humans' linguistic knowledge by first order logic formulae and acquire the proper confidences of the formulae by learning from corpus.

Markov Logic is one of the declarative approaches to structured prediction. Structured prediction usually requires us to consider learning and inference techniques for structured domains in an application-dependent fashion. Because Markov Logic provides us strong learning and inference algorithms in an application-independent manner, we can focus on model constructions for our tasks.

Markov Logic approach has already achieved state-of-the-art results in various NLP tasks such as Entity Resolution (Singla and Domingos, 2006), Information Extraction (Poon and Domingos, 2007), and Semantic Role Labeling (Meza-Ruiz and Riedel, 2009a). We explorer new areas to apply Markov Logic approach by finding effective global constraints based on humans' linguistic knowledges.

1.4 Tasks

Again, our target relations in this thesis are the three types: event-event (E-E), event-argument (E-A), and argument-argument (A-A). We choose the particular relations for each type and illustrate them in Figure 1.3 which can be regarded as an abstract of Figure 1.2.

In Figure 1.3, we have *temporal relation* as event-event relation, *semantic role* for event-argument, and *coreference relation* as argument-argument. For these three types of relations we target the three tasks of relation extraction as follows.

Japanese Event-Argument Relation Extraction

Japanese event-argument relation extraction is a task extracting events (predicates), their argument, and relations (case roles) between them. The case roles are *Ga, Wo* and *Ni*. This work deeply focus on event-argument (E-A) relation.

Most previous work builds separated classifiers corresponding to each case role and independently identified the PA relations, neglecting dependencies (constraints) between two or more PA relations. We propose a method which collectively extracts PA relations by optimizing all argument candidates in a sentence. Our method can jointly consider dependency between multiple PA relations and find the most probable combination of events and their arguments in a sentence. In addition, our model involves new constraints to avoid considering inappropriate candidates for arguments and identify correct PA relations effectively. Compared to the state-of-the-art, our method achieves competitive results without large-scale data.

Biomedical Event-Argument Relation Extraction with Coreference Relation

Biomedical event extraction is a task similar task to Japanese Event-Argument relation extraction. But bio-events, which occur in biomedical documents, have some distinctive characteristics. Arguments of an event may be not only noun that denote physical objects such as protein names but also other events. Role labels are *Theme* and *Cause*. Though recent work has neglected coreference information, we identify coreference relations and propose an approach that exploits them for extracting event-argument relations. We consider this work as a collaboration between relations of event-argument (E-A) and argument-argument (A-A).

This approach has two advantages: it can extract a large number of valuable E-A relations based on the concept of *salience in discourse*; it enables us to identify E-A relations over sentence boundaries (cross-links) using *transitivity* of coreference relations.

Temporal Relation Identification (Temporal Ordering)

Temporal relation identification is a task identifying temporal orders of events or temporal expressions. Though relation labels are mainly *BEFORE, AFTER*, and *OVERLAP*, we have some minor labels *BEFORE-OR-OVERLAP, OVERLAP-OR-AFTER*, and *VAGUE*. In this task, we mainly focus on the relation type of event-event (E-E).

Recent work on temporal relation identification has focused on three types of relations between events: temporal relations between an event and a time expression, between a pair of events and between an event and the document creation time. These types of relations have mostly been identified in isolation by event pairwise comparison. However, this approach neglects logical constraints between temporal relations of different types that we believe to be helpful. We therefore propose a Markov Logic approach that jointly identifies relations of all three relation types simultaneously.

1.5 Thesis Overview

The remainder of this thesis is organized as follows:

Chapter 2: Event Structure and Related Work This chapter mentions the definition of events and explain the differences of events we target in the three tasks. We also introduce the related work for various event structure.

Chapter 3: Preliminaries This chapter describes preliminary techniques for event structure analysis. We make a brief introduction of local classifiers –Support Vector Machines, Log-linear models, and Markov Logic as a probabilistic logic framework. We mainly focus on Markov Logic and introduce the definition, learning and inference algorithms we selected. The proposed Markov Logic Networks for each task are described in Chapters 4, 5, and 6, respectively.

Chapter 4: Japanese Predicate-Argument Relation Extraction This chapter depicts extracting Japanese predicate-argument relation with Markov Logic. We apply our method to Japanese newswire texts and add more qualitative analysis.

Chapter 5: Biomedical Event Extraction We propose coreference based approach for biomedical event extraction. Our work is the first research to exploit coreference relations in biomedical event extraction.

Chapter 6: Temporal Relation Identification In this chapter, we tackle temporal relation identification with Markov Logic. Our novel approach is a global optimization considering multiple temporal relations simultaneously.

Chapter 7: Conclusion This chapter summarizes this thesis and discusses the direction of future work.

Chapter 2

Event Structure and Related Work

In this chapter, we first define the terms, *event*, *predicate*, and *argument* for the three tasks. Then, we describe the concept of *event* mainly focusing on the differences in the three tasks we target.

2.1 Definition

The term *event* refers to a change of state (of situation, of a form of behavior) – the transition from one state to another, usually with reference to a character (agent or patient) or a group of characters. Such characters are called *arguments*. *Predicate* is the most important event which is played by verb or adverb. For instance, the following sentence have a predicate and its arguments, an agent and a patient.

He_{AGT} visited_{PRED} Kyoto_{PAT}

(*AGT*= agent, *PRED*= predicate, *PAT*= patient)

The concept of event has become prominent in recent work on narratology. Event is generally used to help define *narrativity* in terms of the sequentiality inherent to the narrated story. The sequentiality involves changes of state in the represented world and therefore implies the presence of temporality time, which is a constitutive aspect of narration and distinguishes it from other form of discourse such as description or argumentation. Accordingly, such narrative events represent a chronologically ordered sequence of states (Hühn, 2011).

Table 2.1: The Three Types of Events

Type	Sub-type	Example
Japanese-event	predicate	行った (went), 訪問する (visit), 歩く (walk)
	event-noun	影響 (influence), 運転 (driving), 交渉 (negotiation)
bio-event	bind, phosphorylation, up-regulation, adipocyte differentiation	
temporal-event	visited, run, my thesis defense, the giant earthquake	

2.2 Differences in Tasks

In Table 2.1, we summarize the three types of events, *Japanese-events*, *bio-events*, and *temporal-events*.

Japanese Event-Argument Relation Extraction (Chapter 4) deals with events, arguments and the relations between them. We focus on the analysis of verbal and adverbial events so called *predicates* as typical event expressions. Predicates have *arguments* which represent entities that are involved in the activity of predicates such as *agent* or *patient*. Note, predicate-argument structures are often called *semantic-role* and the analysis of them is called *semantic role labeling*. The task in Chapter 4 is a notable instantiation of extracting such arguments.

For example, the sentence

Taro to school went
太郎は$_{NOM}$ 学校へ$_{DAT}$ 行った$_{PRED}$

(Taro went to school.)

(NOM=nominative, DAT=dative, PRED=predicate)

has a predicate –“行った (went)” and its arguments –“太郎は (Taro)” and “学校へ (to school)” as *Ga* (Nominative) and *Ni* (Dative cases), respectively. Though verbs and adverbs play important parts of predicate-argument structures and we actually focus on predicates, there are NPs which have similar structure to predicate-argument. Such NPs are called *event-nouns*. An example sentence for event-noun is

this trade deficit our country's competitiveness influence affect
この 貿易 赤字は$_{NOM}$ 我が国の 競争力に$_{DAT}$ 影響を$_{event}$ 及ぼす

(The trade deficit affects our country's competitiveness.)

in which “影響 (influence)” is an event-noun with two arguments. Such event-nouns are more difficult to extract their arguments than predicates (Komachi et al., 2007).

Japanese event-argument structure for predicates and event-nouns have been annotated in Kyoto Text Corpus (Kawahara et al., 2002) and NAIST Text Corpus (Iida et al., 2007). The annotations of predicate-argument structure in the two corpora are different in dealing with syntactic case alteration. Usually, syntactic cases such as *Ga* or *Wo* directly describe the argument labels (semantic-roles). However, an issue arises in alteration of syntactic cases by syntactic transformations such as passivization and causativization. In the following sentence, we show an example of causativization with *syntactic cases*.[1]

	Hanako	Taro	apple	eat
(1)	花子が	太郎に	リンゴを	食べさせる

Hanako-NOM Taro-DAT apple-ACC eat-CAUSATIVIZED

(Hanako helps Taro eat an apple.)

One way of annotating these predicate-argument structure is

(2) { PRED=食べさせる (eat-causative), NOM=花子 (Hanako), DAT=太郎 (Taro), ACC=リンゴ (apple)}.

This annotation completely follows the syntactic cases and is called *surface* case annotation.

An alternative way of annotation is that first we transform the causativized predicate (e.g. 食べさせる) to *base form* (食べる) then identify its arguments. This way is called *deep* (or *logical*) case annotation. We show deep case annotation of (1) as,

(3) { PRED=食べる (eat), NOM=太郎 (Taro), ACC=リンゴ (apple), EX-NOM=花子 (Hanako) }.

where predicate is not "食べさせる" but "食べる" and the Nominative argument becomes Taro. While Kyoto Text Corpus selected surface case annotation for annotating predicate-argument structure, NAIST Text Corpus utilized deep case annotation for annotating predicate-argument structure.

Moreover, though NAIST Text Corpus is annotated only three major cases, Nominative, Accusative, and Dative, Kyoto Text Corpus has some more cases such as Ablative and Instrumental.

Again, in this task, we define verbal and adverbial events as *predicates*. In addition, we define the set of *predicates* and *event-nouns* as just *events*.

[1]The syntactic cases are just syntactic relations and different from predicate-argument structure

In English, predicate-argument structures for verbs are defined in FrameNet (Baker et al., 1998) and PropBank (Palmer et al., 2005). FrameNet is primarily a lexicographical project which aims to group words into semantic classes so-called *frames*, representations for prototypical situations or states. Each frame provides its set of predicate-argument structures (semantic roles). PropBank has the more practical aim than FrameNet which is to obtain a complete predicate-argument structure annotation of the Penn TreeBank (Marcus et al., 1994). The more detailed of the differences between FrameNet and PropBank is shown in (Ellsworth et al., 2004). For event-noun, Meyers et al. (2004) built the NomBank corpus. Recently, Gerber and Chai (2010) constructed annotations of event-nouns for more difficult cases than NomBank.

Note, English predicate-argument structure analysis, case-frame dictionaries in which predicates proper frames to select their arguments are defined. That is, we can choose a *sense* for each predicate. Because acceptable argument roles differ for each predicate sense, case-frame dictionaries help extract predicate-argument. On the other hand, predicate-argument annotations in Japanese do not have internal case-frame dictionary corresponding to the annotations. Instead, we usually exploits external dictionaries such as Nihongo-Goi-Taikei (Ikehara et al., 1997) or KUCF (Kawahara and Kurohashi, 2006a). Thus, there remains some ambiguities in predicate sense.

Since structure analysis of Japanese event-nouns are challenging task, in Chapter 4, we only focus on predicates – verbal and adverbial events.

Chapter 5 – Biomedical Event Extraction – describes another story of event-argument structure. In biomedical corpus such as GENIA Event Corpus (GEC) (Kim et al., 2008), bio-events have distinctive event structures.[2]

The main task of biomedical event extraction is to find molecular events which involve these entities as their primary participants, *themes* and *cause*. These two types of arguments are different from entities in general domains. Instead of general named entities (e.g. PERSON and ORGANIZATION), bio-events take named entities of the protein, gene and RNA types as their arguments. In addition, biomedical corpus has large numbers of nominal events and an event can take the other events as arguments. Bio-events have *event-types* defined based on Gene Ontology which is a project aiming at standardizing the representation of gene attributes across species and databases.[3] In terms of event-argument structure, event-type can be regarded as a similar idea to case-frame which decide acceptable arguments for each event. We show an example

[2]Note that the bio-events in this thesis are limited events annotated in GEC.

[3]`http://www.geneontology.org/`

sentence of Figure 5.2 as follows:

Pos_reg *Pos_reg* *Binding*
TPA [induction] [increases] the [binding] of [AP-1 factors] to [this element].
event *event* *event* *protein* *protein*

which has three events “induction”, “increases”, and “binding” and two proteins “AP-1 factors” and ‘this element”. Each event has an event-type such as *Positive_Regulation* and *Binding*. As stated before, the roles between events and their arguments are two types, *theme* and *cause*. This example sentence is in BioNLP’09 (Kim et al., 2009) in which we analyze event-argument structure we stated above. Event-argument structure of bio-event is domain specific but the variety of proteins and nominal events make analysis difficult.

In Chapter 6 we focus on the aspect of chronological order of events. Different from the other two tasks in Chapters 4 and 5, the problem we target is not extracting the attributes of *who* or *what* for events but the information *when* they happen. Since an event is a change of state, generally all events have temporal information. Here, we call events with temporal information *temporal-events*. Temporal-events do not necessarily have arguments such as agents or patients. In addition to predicates and event-noun of the above, temporal-events include much more varieties of nominal events. Many historical occurrence should become temporal-events. For example, “the giant earthquake”, “World War II ” can be temporal-events because they have temporal information when they occur. We have popular temporal tagged corpus TimeBank (Pustejovsky et al., 2003a) and TempEval (Verhagen et al., 2007). In them, there are several types of temporal relations such as relations between event and time expression (e.g. July 7th) or relations between two events. Temporal relation identification is a task to identify these relations (temporal order) based on 13 or 6 classes (e.g. *Before*, *Overlap*).

Note that, in this section, we explicitly describe each type of events such as “Japanese-event” and “bio-events” but, in the remainder of this thesis, we call them just *events*.

2.3 Related Work for Event

In this section, we make a brief introduction of related work. We describe researches related to the three types of relations in Figure 1.3 or Markov Logic approach. The detailed approach or results will be described in each of Chapters 4, 5, and 6.

First, in Japanese Predicate-Argument Structure (PAS) analysis, Taira et al. (2008) and Imamura et al. (2009) tackled PAS extraction on the NAIST Text Corpus. While Imamura et.al., focused on verbal and adverbial predicates, Taira et.al., included *event-noun* in their targets. Both works have their points: Taira's model is strong in dative cases and Imamura's work got outperformed results in nominative and accusative cases. For extraction of predicate-argument relations over sentence boundaries, there is much room for improvement. Markov Logic approach to extract Japanese PAS has not proposed yet.

Second, in biomedical event extraction, we mention the works in BioNLP'09 Shared Task (Kim et al., 2009) which is one of the most popular shared tasks for biomedical event extraction. In this shared task, 24 teams submitted final results. BioNLP'09 has three tasks but most teams tried only Task 1 –extracting basic event structure. Björne et al. (2009) won the competition using very simple method with SVM classifiers. Riedel et al. (2009) proposed a novel Markov Logic approach to biomedical event extraction. On the data of BioNLP'09, Poon and Vanderwende (2010) proposed a new Markov Logic model by implementing the features (Björne et al., 2009) used and achieved competitive results with Björne et.al. Our model for this task is also applying Markov Logic but we involves coreference informations.

Thirdly, let us focus on temporal relations. Temporal relations are traditionally defined as 13 types by Allen's temporal logic (Allen, 1983). Pustejovsky et al. (2003b) built a temporal annotation format called TimeML. TimeML has the formats to annotate event expressions, temporal expressions, and temporal relations. The format for temporal relations are 11 types based on Allen's temporal logic. TimeBank Corpus (Pustejovsky et al., 2003a) is the most popular corpus with temporal annotations which is annotated by TimeML format. Boguraev and Ando (2005) first proposed a machine learning approach to temporal relation identification on TimeBank Corpus. After the TimeBank Corpus, same as biomedical event extraction, shared tasks are held. TempEval shared task,[4] TempEval-2,[5] and TempEval-3 [6] are held in 2007, 2010, and 2011, respectively. The data for these shared tasks were newly constructed. Temporal relations in TempEvals are simplified 6 types. Though many approaches were proposed in TempEvals, global features were not exploited enough.

[4] http://www.timeml.org/tempeval/

[5] http://www.timeml.org/tempeval2/

[6] http://www.cs.york.ac.uk/semeval/proposal-1.html

Chapter 3

Preliminaries

This section describes the prelimnaries of our work. It is a brief introduction of machine learning techniques and probabilistic logic framework.

3.1 Support Vector Machines

Support Vector Machines (SVMs) are supervised machine learning models for binary classification proposed by Vapnik (1995).

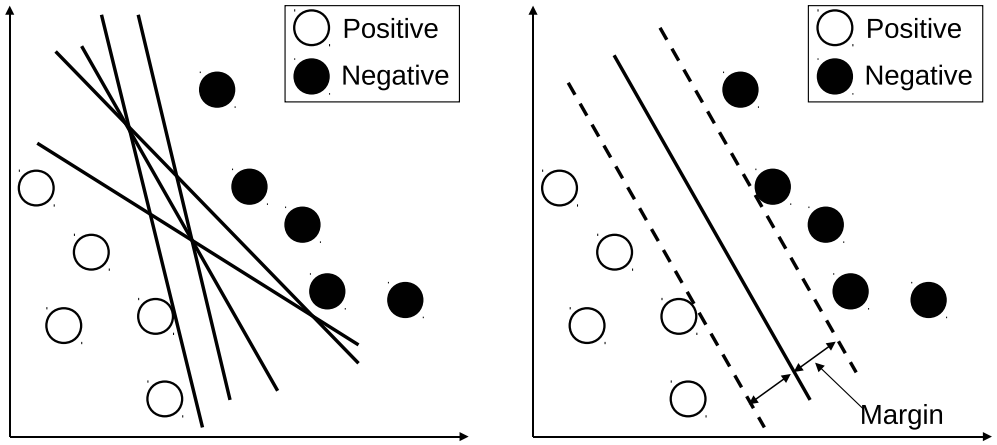
Given a set of training example $\mathcal{D} = \left\{(x_i, y_i)|x_i \in \mathbf{R}^d, y_i \in \{-1, 1\}, 1 \leq i \leq n\right\}$, suppose the following hyperplane:

$$\mathbf{w} \cdot \mathbf{x} + b = 0, \qquad \mathbf{w} \in \mathbf{R}^d, b \in \mathbf{R}, \tag{3.1}$$

which separates the training data into two classes such that

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \geq 0. \tag{3.2}$$

Though there are many possible hyperplanes (Figure 3.1(a)), SVMs find the optimal hyperplane that maximizes the margin (Figure 3.1(b)). Such a hyperplane has the minimum expected test errors and can be solved by quadratic programming. The inequality 3.2 must hold for the nearest examples. Those nearest examples form two margin-boundary hyperplanes formed by the nearest examples of positive and negative, respectively. Let λ be the distance between two margin-boundary hyperplanes, and $\bar{\mathbf{x}}$ be a vector on the margin-boundary hyperplane formed by the nearest negative examples. Then, the following equations hold:



(a) Possible Separating Hyperplanes (b) Optimal Separating Hyperplane

Figure 3.1: Maximizing the Margin of Support Vector Machine

$$-1 \times (\bar{\mathbf{x}} \cdot \mathbf{w} + b) - 1 = 0 \tag{3.3}$$

$$1 \times ((\bar{\mathbf{x}} + \lambda \mathbf{w}/|\mathbf{w}|) \cdot \mathbf{w} + b) - 1 = 0 \tag{3.4}$$

The margin is the half of the distance λ and computed as

$$\frac{\lambda}{2} = \frac{1}{|\mathbf{w}|} \tag{3.5}$$

Thus, maximizing the margin is equivalent to minimizing the norm of $\mathbf{w}$. This problem is simply formulated as:

$$\begin{aligned} \min \quad & \frac{1}{2}|\mathbf{w}|^2, \\ s.t. \quad & \forall i, \quad y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \geq 0 \end{aligned} \tag{3.6}$$

which represents *hard-margin* SVMs where the given data is linearly separable.

On the other hand, *soft-margin* SVMs introduces the so-called slack variables which enables the linearly non-separable problem to be solved. This problem is formulated as

$$\begin{aligned} \min \quad & \frac{1}{2}|\mathbf{w}|^2 + C \sum_i \xi_i, \\ s.t. \quad & \forall i, \quad y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \xi \geq 0. \\ & \forall i, \quad \xi_i \geq 0 \end{aligned} \tag{3.7}$$

where $\xi_i(\forall i)$ are slack variables and C is a user-given constant. The intuition behind this formulation is that as few examples as possible are allowed to penetrate into the margin or event into the other side of the hyperplane. The parameter C controls the trade-off between the margin and the size of the slack variables. C is a penalty factor which allows us to trade off training error and model complexity. A small value for C will increase the number of training errors, while a large C will lead to a behavior similar to that of a hard-margin SVM.

Given a test example $\mathbf{x}$, its label y is decided by the sign of the discriminant function $f(\mathbf{x})$:

$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b \tag{3.8}$$

$$y = \text{sgn}(f(\mathbf{x})). \tag{3.9}$$

Another way of deal with linearly non-separable cases is *kernel method*. In the kernel method, feature vectors are mapped into a higher dimensional space by a nonlinear function $\Phi(\mathbf{x})$ and linearly separated there. Since all examples still appear in forms of inner products, what we have to do is just calculating the inner product of two examples in the higher dimensional space. Those values may be calculated in $\mathbf{R}^d$ without mapping into the higher dimensional space by the following function $K(\mathbf{x}_i, \mathbf{x}_j)$,

$$K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) \tag{3.10}$$

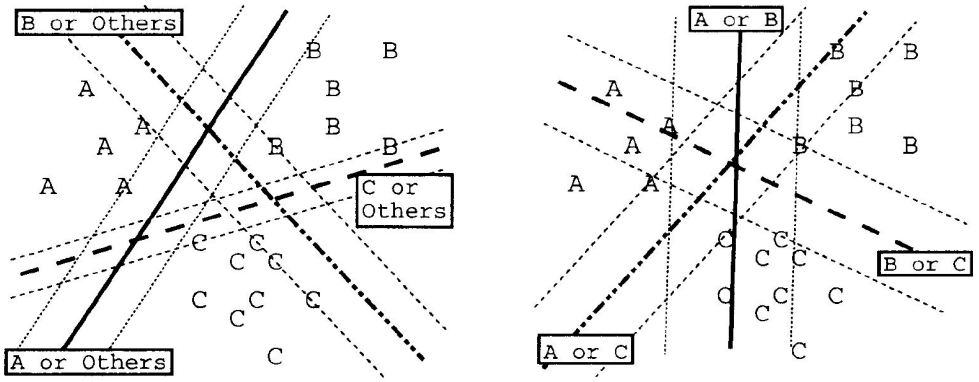
The functions that conduct such calculation are called *kernel functions*. For example, a popular kernel function called polynomial kernel is represented as:

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^l \tag{3.11}$$

which virtually maps the original input space into a higher dimensional space where all combinations of up to d features are taken into account.

We can also solve multi-class problems as a variation of SVM (Hastie and Tibshirani, 1998; Weston and Watkins, 1998). In Figure 3.2, we illustrate two simple heuristic strategies for dealing a multi-class classification by SVMs.

One vs Rest classification (Figure 3.2(a)) creates N classifiers for N-class classification task. As the figure, if there are three classes (A,B,and C), we need three classifiers. We classify them independently and decide the most confident class as the answer for a data to classify. In order to measure the confidence of classes, we compare the distances from the hyperplanes of each class.



(a) One vs Rest classification (b) Pairwise classification

Figure 3.2: Multi-class Support Vector Machine

Pairwise classification (Figure 3.2(b)) requires ${}_NC_2$ classifiers for N-class classification tasks. After solving each classifier separately, we decide an answer by majority voting for a data.

Although these two strategies work well for our task, we exploit a more sophisticated package; SVM^{struct}.[1] It supports multi-class mode and its performance is generally better than One vs Rest and Pairwise because of the global optimization to multi parameters. In our experiment, we use only this package and do not consider heuristic methods any more. SVMs have achieved high performance in various tasks. We consider the approach using SVMs as baseline methods for each task.

3.2 Log-Linear Model

Log-Linear Model (LLM) has been popular and widely used in NLP classification tasks (Berger et al., 1996; Ratnaparkhi, 1998; Smith, 2004). Log-linear models assign conditional probabilities to observation/label pairs $(\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{Y}$,

$$P(\mathbf{y}|\mathbf{x}) = \frac{\exp\Big(\mathbf{w} \cdot f(\mathbf{x}, \mathbf{y})\Big)}{\sum_{\mathbf{y}' \in \mathcal{Y}} \exp\Big(\mathbf{w} \cdot f(\mathbf{x}, \mathbf{y}')\Big)} \tag{3.12}$$

where $\mathbf{w}$ is a weight vector and $f(\mathbf{x}, \mathbf{y})$ is a function that maps pairs $(\mathbf{x}, \mathbf{y})$ to a feature vector. The ith feature value is given by $f_i(\mathbf{x}, \mathbf{y})$ and the corresponding ith weight

[1] `http://svmlight.joachims.org/svm_struct.html`

value is represented as $\mathbf{w}_i$. Given training examples $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{y}'_1), \ldots, (\mathbf{x}_n, \mathbf{y}'_n)\}$, we can estimate maximum likelihood of such a model by solving the following optimization problem.

$$\begin{aligned} \mathbf{w}^*_{ML} &= \arg\max_{\mathbf{w}} \prod_{i=1}^{n} P(\mathbf{y}'_i|\mathbf{x}_i) \\ &= \arg\max_{\mathbf{w}} \sum_{i=1}^{n} \log P(\mathbf{y}'_i|\mathbf{x}_i) \end{aligned} \tag{3.13}$$

Commonly maximizing conditional log-likelihood often causes parameters overfitting if we do not have enough training examples for many parameters. In order to alleviate parameters overffiting , Equation 3.13 is added regularization term $r(\mathbf{w})$ for each feature:

$$\mathbf{w}^*_{MAP} = \arg\max_{\mathbf{w}} \sum_{i=1}^{n} \log P(\mathbf{y}'_i|\mathbf{x}_i) - C \cdot r(\mathbf{w}) \tag{3.14}$$

where C is a tuning parameter: under low C models tend to fit data and high C strengthens the effect of the regularization.

There are two popular regularization ways called L_1 and L_2. L_1 regularization uses $r(\mathbf{w}) = \sum_{i=1}^{n} |w_i|$. L_2 regularization exploits $r(\mathbf{w}) = \sum_{i=1}^{n} w_i^2$. Note that applying *Laplacian* and *Gaussian* distributions to prior distribution correspond to L_1 and L_2 regularizations, respectively. We now optimize the posterior probability $P(\mathcal{D}, \mathbf{w}) = P(\mathcal{D}|\mathbf{w})P(\mathbf{w})$ and it is called *maximum a posteriori (MAP)* estimation.

This objective functions represented as Equations 3.13 and 3.14 are concave and can therefore be optimized using numerical optimization procedures such as L-BFGS (Liu et al., 1989) or conjugate gradient (Hestenes and Stiefel, 1952; Daumé III, 2004).

3.3 Markov Logic

Markov Logic was proposed by Richardson and Domingos (2006) as a powerful framework for Statistical Relational Learning (SRL) (Ng and Subrahmanian, 1992; Koller, 1999).

SRL is a newly emerging research area from the requests to deal with real-world data more precisely and efficiently. In general, real-world datasets have characteristics of both uncertainty and complex relational structure. Statistical learning focuses on

the former and relational learning on the latter. SRL is at the intersection of statistical learning and relational learning, namely it has the power of both. Because of its flexibility, SRL has grown rapidly in recent years in various fields.

Markov Logic is an expressive representation of SRL that generalizes both full first-order logic and Markov networks (Richardson and Domingos, 2006; Domingos and Lowd, 2009). Markov Logic is often called probabilistic logic or weighted logic, because it is a combinational framework of probabilistic model and logical formula. Markov Logic successfully works several tasks such as Entity Resolutions (Singla and Domingos, 2006), Information Extraction (Poon and Domingos, 2007), and Web data mining (Wu and Weld, 2008). Riedel and Meza-Ruiz (2008; Meza-Ruiz and Riedel (2009a) successfully used MLNs for semantic role labeling.

In the following subsections, we make a brief introduction of First Order Logic and Markov Networks, and then describe Markov Logic, its definition and algorithms of learning and inference.

3.3.1 First Order Logic

A first-order knowledge base (KB) is a set of formulae in first order logic (Genesereth and Nilsson, 1987). Formulae includes four types of symbols: predicate, function, constant, and variable. In this section, we sometimes call predicates *logical predicates* in order to distinguish them to linguistic ones.

(logical) predicate represents relations among objects in the domain or attributes of objects (e.g. *See, Visit, Talkative*)

function represents mappings from tuples of objects to objects (e.g. *MotherOf*)

constant represents objects in the domain of interest (e.g., people: *Taro, Hanako*, place: *Tokyo, Library*, etc.)

variable ranges over the objects in the domain (e.g., people: *X, Y*, place: *A, B*)

Variables and constants are typed such as *people* or *place*, in which case variables range over objects of the corresponding types. That is, variable *X, Y* might range over people (e.g., *Taro, Hanako*).

Arbitrary expression representing an object in the domain is called *term* which can be a constant a variable, or a function applied to a tuple of terms (e.g., *Taro, X, MotherOf(X,Y)*). An atomic formula or atom is a logical predicate symbol applied to a tuple

of terms (e.g. $See(Taro, MotherOf(Hanako))$). Formulae are recursively constructed from atomic formulae using logical connectives and quantifiers. If F_1 and F_2 are formulae, the following are also formulae:

- $(\neg F_1)$ which is true iff F_1 is false (negation)
- $(F_1 \wedge F_2)$ which is true iff both F_1 and F_2 are true (conjunction)
- $(F_1 \vee F_2)$ which is true iff F_1 or F_2 is true (disjunction)
- $(F_1 \Rightarrow F_2)$ which is true iff F_1 is false or F_2 is true (implication)
- $(F_1 \Leftrightarrow F_2)$ which is true iff F_1 and F_2 have same truth value (equivalence)
- $(\forall x.F_1)$ which is true iff F_1 is true for every object x in the domain (universal quantification)
- $(\exists x.F_1)$ which is true iff F_1 is true for at least one object x in the domain (existential quantification)

A *positive literal* is an atomic formula; a *negative literal* is negated atomic formula. A *ground term* is a term containing no variable (instantiated with constants). A *ground atom* or *ground logical predicate* is an atomic formula all of whose arguments are ground terms. A *grounding* is an element grounded from first-order materials (predicate, formula, Markov network, etc.) in which all predicates contain no variable. For instance, a grounding of a formula is a ground atom. A *possible world* assigns a truth value to each possible ground atom.

A formula is *satisfiable* iff there exists at least one world in which the formula is true. The basic inference problem in first order logic is to determine whether a knowledge base (KB) entails a formula F. That is if F is true in all worlds where KB is true ($KB \models F$).

Knowledge bases are often constructed using a restricted subset of first order logic with more desirable properties because inference in first-order logic is only semi-decidable. The most widely used restriction is to *Horn clauses*, which are clauses containing at most one positive literal. The Prolog programming language is based on Horn clause logic (Lloyd, 1987). Prolog programs can be learned from databases by searching for Horn clauses that (approximately) hold in the data; this is studied in the field of inductive logic programming (Lavrac and Dzeroski, 1994).

3.3.2 Markov Networks

Formally, a Markov Network (Markov random field) can be defined as follows:

Definition 2.1 A Markov Network M is a pair (G, Φ) where

- G is an undirected graph $(\mathbf{Y}, E)$ where the vertices $\mathbf{Y} = (Y_i)_i$ are a family of random variables and each edge (Y_i, Y_j) represents a correlation between Y_i and Y_j.
- Φ is a set of non-negative potential functions $(\phi_k)_k$ where k is the kth clique in G and ϕ_k has k as its domain.

Let $\mathbf{y}_{\{k\}}$ be the state of the kth clique (i.e., the state of the variables that appear in that clique). The joint distribution represented by a Markov Network is given by

$$P(\mathbf{Y} = \mathbf{y}) = \frac{1}{Z} \prod_k \phi_k(\mathbf{y}_{\{k\}}) \tag{3.15}$$

where Z is a normalization constant (the so-called partition function):

$$Z = \sum_{\mathbf{y} \in \mathbf{Y}} \prod_k \phi_k(\mathbf{y}_{\{k\}}) \tag{3.16}$$

Dividing by Z guarantees that summing over all possible assignments yields 1.

Markov Networks are often conveniently represented as *log-linear* models we stated in Section 3.2, with each clique potential replaced by an exponentiated weighted sum of features of the state, leading to

$$P(\mathbf{Y} = \mathbf{y}) = \frac{1}{Z} \exp\left(\sum_i w_i f_i(\mathbf{y}) \right) \tag{3.17}$$

where each f_i is a real valued $feature$ function over $\mathbf{y}$ and w_i is its associated weight. In this thesis, we will focus on binary features, $f_i(\mathbf{y}) \in \{0, 1\}$. This representation is exponential in the size of the cliques. However, in most cases feature functions will only consider a few subcomponents of all possible cliques.

Inference in Markov Networks is #P-complete (Roth, 1996). The most widely used method for approximate inference in Markov Networks is Markov Chain Monte Carlo (MCMC) (Gilks and Spiegelhalter, 1996), and in particular Gibbs sampling, which

proceeds by sampling each variable in turn given its Markov blanket.[2] Marginal probabilities are computed by counting over these samples; conditional probabilities are computed by running the Gibbs sampler with the conditioning variables clamped to their given values. The details of MAP inference and learning algorithms are covered by the next subsection.

3.3.3 Definition of Markov Logic

Markov Logic combines first order logic and Markov networks. A traditional first order knowledge base (KB) can be seen as a set of hard constraints on the set of possible worlds: if a world violates even one formula, the world is impossible.

The basic idea of Markov Logic is to soften these too strong constraints: when a world violates one formula in the KB, it is less probable with some penalty. The remarkable point is that the penalized world is only less probable but does not have zero probability.

Thus Markov Logic describes a knowledge base as a set of weighted formulae (or first order features). This set of weighted formulae is referred to as a Markov Logic Network (MLN) and defines a log-linear probability distribution over possible worlds.

DEFINITION A Markov Logic Network L is a set of pairs (ϕ_i, w_i), where ϕ_i is a formula in first-order logic and w_i is a real number. Together with a finite set of constants $C = \{c_1, c_2, \ldots, c_{|C|}\}$, it defines a Markov network $M_{L,C}$ as follows:

1. $M_{L,C}$ contains one binary node for each possible grounding of each logical predicate appearing in L. The value of the node is 1 if the ground atom is true, and 0 otherwise.

2. $M_{L,C}$ contains one feature f for each possible grounding of each formula ϕ_i in L.

$$f_{\mathbf{c}}^{\phi_i}(\mathbf{y}) = \begin{cases} 1 & \text{if } \models_y \phi_i[\mathbf{c}] \\ 0 & \text{otherwise} \end{cases} \tag{3.18}$$

   where $\phi_i[\mathbf{c}]$ is the ground formula we create by replacing each free variable in ϕ_i with the corresponding set of constants $\mathbf{c}$ and recursively expanding each existential quantification with a corresponding disjunction and each universal quantification with a corresponding conjunction. The value of this feature is 1 if the

[2]The Markov blanket of a node is simply the nodes' neighbors in the graph.

ground formula is true, and 0 otherwise. The weight of the feature is w_i associated with ϕ_i in L.

An MLN can be viewed as a template for constructing Markov Networks. Given different sets of constants, it will produce different networks. We call each of these networks a *Ground Markov Network* to distinguish it from the first order MLN. From the Definition and Equations 3.15-3.17, the probability distribution over possible world $\mathbf{y}$ specified by the ground Markov Network $M_{L,C}$ is given by

$$P(\mathbf{Y} = \mathbf{y}) = \frac{1}{Z} \exp\left(\sum_i w_i n_i(\mathbf{y})\right) = \frac{1}{Z} \exp\left(\sum_j w_j g_j(\mathbf{y})\right) \tag{3.19}$$

where $n_i(\mathbf{y})$ is the number of true groundings of ϕ_i in $\mathbf{y}$, $g_j(\mathbf{y})$ corresponds to a ground clause and works as a binary feature of $f_i(\mathbf{y})$ in Equation 3.17. Therefore, $g_j(\mathbf{y}) = 1$ if the jth ground clause is true in the data and 0 otherwise.

More precisely, suppose $M = \{\phi_i, w_i\}$ be a Markov Logic Network again, then Equation 3.19 is represented as

$$P(\mathbf{Y} = \mathbf{y}) = \frac{1}{Z} \exp\left(\sum_{(\phi_i, w_i) \in M} w_i \cdot \sum_{\mathbf{c} \in C^{\phi_i}} f_{\mathbf{c}}^{\phi_i}(\mathbf{y})\right) \tag{3.20}$$

where C^{ϕ_i} contains all constants of the free variables in ϕ_i. $f_{\mathbf{c}}^{\phi_i}$ is a feature function that returns 1 if in the possible world $\mathbf{y}$ the *ground formula* we get by replacing the free variables in ϕ_i with the constants in $\mathbf{c}$ is true, and 0 otherwise.

A possible world becomes more likely the more groundings of formulae with positive weight are true and the more groundings with negative weight are false. More precisely, assume that there are two possible worlds $\mathbf{y}_1$ and $\mathbf{y}_2$ that only differ in one grounding of one formula ϕ_i with weight w_i which holds in $\mathbf{y}_1$ but does not hold in $\mathbf{y}_2$. Then the probability of the possible world $\mathbf{y}_1$ is $exp(w_i)$ times higher than the probability of $\mathbf{y}_2$.

Finally, let us see a famous example of small Markov Logic Network (MLN). Table 3.1 is an MLN example taken from (Richardson and Domingos, 2006). The first and second columns of this table show a simple knowledge base and its conversion to clausal form. Note that, these formulae may be typically true in the real world but not always true. In most domains it is very difficult to come up with non-trivial formulae that are always true, and such formula capture only a fraction of the relevant knowledge. Thus, despite its expressiveness, pure first-order logic has limited applicability

Table 3.1: Example of a first-order knowledge base and corresponding MLN Fr() is short for Friends(), Sm() for Smokes(), and Ca() for Cancer()

First-Order Logic	Clausal Form	Weight
"Smoking causes cancer." $\forall x \text{Sm}(x) \Rightarrow \text{Ca}(x)$	$\neg\text{Sm}(x) \vee \text{Ca}(x)$	1.5
"If two people are friends and one smokes, then so does the other." $\forall x \forall y \text{Fr}(x,y) \wedge \text{Sm}(x) \Rightarrow \text{Sm}(y)$	$\neg\text{Fr}(x,y) \vee \neg\text{Sm}(x) \vee \text{Sm}(y)$	1.1

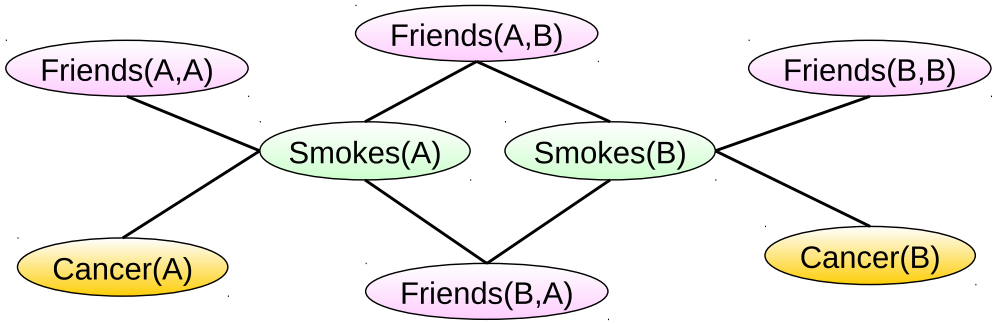


Figure 3.3: Graphical Structure of MLN in Table 3.1

to practical AI problems. But, MLN has the third column, *weight* of each formula which corresponds to Markov network.

Figure 3.3 shows a graphical structure of a ground Markov Network, which is constructed by applying constants A and B to the MLN in Table 3.1. Each node in this graph is a ground atom. There is an edges between each pair of nodes (predicates) iff there is at least one grounding of one formula in which the two ground atoms appear together. MLN can now be used to infer the probability that "A and B are friends given their smoking habits", the probability that "B has cancer given his friendship with A and whether she has cancer", etc.

3.3.4 Inference of Markov Logic

Markov Logic applies characteristic methods for inference. The inference of Markov Logic Networks (MLNs) is finding the most probable state of the world given evidences and this is known as MAP inference. In MLNs, this inference becomes a problem to find the truth assignment that maximizes the sum of weights of satisfied

clauses. We are given a MLN M and given ground atoms $(\mathbf{x}_{P(\mathbf{c})})_{p\in O,\mathbf{c}\in C}$ for a set of *observed* predicates O, and a set of constraints C. What to solve is to find the set of *hidden* ground atoms $\hat{\mathbf{y}} \in \mathcal{Y}_{H,C}$ for a set of remaining logical predicates H with *maximum a posteriori* (MAP) probability

$$\hat{\mathbf{y}} = \underset{\mathbf{y}\in\mathcal{Y}_{H,C}}{\arg\max}\ P(\mathbf{y}|\mathbf{x}) = \underset{\mathbf{y}\in\mathcal{Y}_{H,C}}{\arg\max}\ s(\mathbf{y},\mathbf{x}) \tag{3.21}$$

where

$$s(\mathbf{y},\mathbf{x}) = \sum_{(\phi,w)\in M} w \sum_{\mathbf{c}\in C^{\phi}} f_{\mathbf{c}}^{\phi}(\mathbf{y},\mathbf{x}) \tag{3.22}$$

is considered as a linear *scoring* function that evaluates the goodness of a problem solution pair $(\mathbf{x},\mathbf{y})$.

MaxWalkSAT

In order to solve this problem, the original method of MLNs inference was MaxWalkSAT (MWS) (Richardson and Domingos, 2006). It is a weighted variant of the WalkSAT (Selman et al., 1993). WalkSAT is a local search algorithm to solve boolean satisfiability problems. However, MLNs inference requires to take care of weighted clauses, MWS evaluates the cost of unsatisfied clauses with their weights based on Equation 3.22.

We show the flow of MWS in Algorithm 1. It starts by assigning a random state to all ground atoms and get forwards by repeatedly picking a random unsatisfied ground clause. MWS has two kinds of steps in the later processes. The *random* step is randomly picking an unsatisfied ground atom in the selected unsatisfied clause with a probability q. The *greedy* step is choosing the ground atom which gets the largest increase of $s(\mathbf{x},\mathbf{y})$. After a ground atom is selected, then flipped the state 1 or 0. These processes are repeated until the fixed number of flips is reached.

Though MWS has been successfully used for various weighted satisfiability problems, the inference with MWS can be inaccurate because of using an approximate Random Walk method.

Integer Linear Programming

Integer Linear Programming (ILP) has been used for several MAP inference tasks (Roth and tau. Yih, 2005; Riedel and Clarke, 2006; Clarke and Lapata, 2007). In contrast to

Algorithm 1 MaxWalkSAT ($M, n_{restarts}, n_{flips}$)

Require: A Markov Network M with set of clausal features, the number of restarts $n_{restarts}$ and the number of flips n_{flips}

```
r ← 0
while r ≤ n_restarts do
    y ← random
    i ← 0
    while i ≤ n_flips do
        c ← randomUnsatisfiedClause(M, y)
        u ← random(0, 1)
        if u < p then
            for a' ∈ c do
                s_a' ← deltaScore(M, a', y)
            end for
            a ← max_{a'∈c} s_a'
        else
            a ← randomVariable(c)
        end if
        y_a ← 1 − y_a
        i ← i + 1
    end while
    r ← r + 1
end while
return y
```

MWS, it can solve the MAP problem exactly: if an ILP solver terminates, the returned assignment will be the true optimal solution. However, the disadvantage of ILP-based inference are memory and runtime requirements. For larger problems applying ILP often becomes infeasible.

Though ILP solvers can be implemented in several ways, we deal with them as black boxes. We choose a free ILP solver and do not explain how it finds optimal solutions for ILP problems. This is one of the advantages that treat a MAP problem as an ILP problem because it helps to solve complex MAP tasks with minimal engineering effort.

An Integer Linear Program (Winston and Venkataramanan, 2003) solves a con-

strained optimization problem as follows:

$$\begin{aligned} &\arg\max_{\mathbf{x}\in\mathbb{N}^{\kappa}} \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ &\forall i \in \{1,\ldots,m\} : {\alpha_{\mathbf{i}}}^{\mathrm{T}}\mathbf{x} \leq \beta_i \end{aligned} \tag{3.23}$$

where $\mathbf{c} \in \mathbb{R}^n$ is a *cost* vector and the $\alpha_{\mathbf{i}} \in \mathbb{R}^n$ and β_i construct a linear inequality constraints. ILP problem has various variation but the ILP problem we are interested in is 0-1 (Boolean) Linear Programs in which variables $\mathbf{x}$ take values only 0 or 1. In other words,

$$\begin{aligned} &\arg\max_{\mathbf{x}\in\{0,1\}^n} \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ &\forall i \in \{1,\ldots,m\} : {\alpha_{\mathbf{i}}}^{\mathrm{T}}\mathbf{x} \leq \beta_i \quad . \end{aligned} \tag{3.24}$$

We can map MAP inference problem to this type of ILP problem based on Riedel's work (Riedel, 2008). This is a mapping for binary Markov Networks represented with log-linear features. Here we see log-linear representation of the MAP problem:

$$\arg\max_{\mathbf{y}} \frac{1}{Z} \exp\left(\sum_i f_i(\mathbf{y},\mathbf{x}) \cdot \theta_i\right) \tag{3.25}$$

where $\mathbf{y}$ is a binary vector and every f_i is a binary feature function over $\mathbf{y}$ represented as a propositional formula such as

$$f_1(\mathbf{y},\mathbf{x}) = \begin{cases} 1 & \text{if} \neg y_1 \vee x_1 \\ 0 & \text{otherwise} \end{cases} \tag{3.26}$$

Let us start by replacing each feature function application $f_i(\mathbf{y},\mathbf{x})$ in equation 3.25 with a binary auxiliary variable λ_i and constrain $f_i(\mathbf{y},\mathbf{x})$ and λ_i to be equal. This leads to the optimization problem

$$\begin{aligned} \arg\max_{\mathbf{y}} \quad & \sum_i \theta_i \lambda_i \\ s.t. \quad & \lambda_i = f_i(\mathbf{y},\mathbf{x}), i = 1,\ldots,n \end{aligned} \tag{3.27}$$

with linear objective function under a set of constraints.

In order to turn this into an ILP, we need to transform each equality constraint into a set of linear constraints over $\mathbf{y}$ and the auxiliary variables $(\lambda_i)_i$. The constraint $\lambda_i = f_i(\mathbf{y},\mathbf{x})$ can be transformed to linear constraint as follows:

1. Mapping each constraint to a logical equivalence of the auxiliary variable and the logical formula, the feature is based on, such as

$$\lambda_1 \Leftrightarrow \neg y_1 \vee x_1 \tag{3.28}$$

2. Replacing the x_i variables by their values in $\mathbf{x}$ (i.e. either *true* or *false*), leading to formulae such as

$$\lambda_1 \Leftrightarrow \neg y_1 \vee false \tag{3.29}$$

3. Transforming the logical equivalence into Conjunctive Normal Form (while eliminating disjunctions that are always true and literals that are always false), as in

$$(\neg\lambda_1 \vee \neg y_1) \wedge (\lambda_1 \vee y_1) \tag{3.30}$$

4. Replacing each disjunction by a linear constraint (Williams, 1999), for example

$$\begin{aligned} -1 \cdot \lambda_1 - 1 \cdot y_1 &\geq -1 \\ 1 \cdot \lambda_1 + 1 \cdot y_1 &\geq 1 \end{aligned} \tag{3.31}$$

Using this representation the number of variables can be reduced significantly. If n is the number of binary variables and m is the number of features, this mapping creates $n + m$ binary variables, one for each node and one for each feature. Thus even if the number of nodes included in potential/feature is high, the number of ILP variables remains low because there is exactly one ILP variable for each feature. For example, even in the case of a clique with 30 binary variables, we only need one variable to represent the value of the potential.

Cutting Plane Inference

Riedel (2008) proposed another efficient inference method named Cutting Plane Inference (CPI). The algorithm of CPI is a variant of the Cutting Plane approach from Operations Research (Dantzig et al., 1954). The advantage of Cutting Plane approach is that it is able to solve large scale constrained optimisation problems by only considering a subset of constraints. Since many NLP works use so many features, running inference by using the full groundings of a MLNs can often be slow and the output of it can be inaccurate when using MaxWalkSAT. Actually CPI addresses these problems.

Instead of searching for unsatisfied constraints, CPI searches for the ground formulae *not maximally satisfied* in the world $\mathbf{y}'$. In other words, for each formula ϕ and a given $(\mathbf{y}', \mathbf{x})$, CPI is looking for all tuples, $Separate(\phi, w, \mathbf{y}, \mathbf{x}) \subseteq C^{\phi}$, for which

$$w \cdot f_{\mathbf{c}}^{\phi}(\mathbf{y}', \mathbf{x}) < \max_{\mathbf{y} \in \mathbf{Y}_H, C} w \cdot f_{\mathbf{c}}^{\phi}(\mathbf{y}, \mathbf{x}) \tag{3.32}$$

In the terminology of the Cutting Plane approach, this step is regarded as *separation*: it finds a set of constraints that separates feasible solutions from infeasible solutions. It can help to separate possible worlds with high score from those with low score.

CPI defines a partial grounding $\mathbf{G} = (G_{\phi})_{(\phi,w)} \in M$ with $G_{\phi} \subseteq C^{\phi}$ that maps each formula ϕ_i to a set of tuples grounded it with. A partial groundings induces a *partial score*

$$s_G(\mathbf{y}, \mathbf{x}) = \sum_{(\phi,w) \in M} w \sum_{c \in G_{\phi}} f_c^{\phi}(\mathbf{y}, \mathbf{x}). \tag{3.33}$$

The flow of CPI is described in Algorithm 2. In each iteration i, CPI checks and updates a partial grounding $\mathbf{G}^i$. Initially $\mathbf{G}^0$ is filled with a small number of groundings. Usually, $\mathbf{G}^0$ has all groundings of formulae which only contain one hidden predicate. In this case maximising $s_{\mathbf{G}^0}$ is easy because the hidden variables do not interact and often gives a very good first guess.

In step 5, CPI finds a solution $\mathbf{y}$ that maximises the partial score $s_{\mathbf{G}^{i-1}}$. To find this solution, CPI needs a base solver. ILP solver[3] is the default option because of its exactness, effectiveness, and declarative nature.

In steps 9 and 10, it finds the ground formulae which are not maximally satisfied in the current solution $\mathbf{y}$ and add them to the current partial grounding. It terminates if no more new ground formulae are found or a maximum number of iterations is reached. This process calculates one solution $\mathbf{y}$ in each iteration. The final result is the solution $\mathbf{y}$ with highest score.

In general, event structure analysis requires to consider many possible candidates of event and arguments simultaneously. Therefore, the analysis is computationally very hard and the speeding up by CPI is an essential technique to build our Markov Logic models. We exploit an Markov Logic Engine which implements CPI and apply this algorithm to the each task we tackle in Chapters 4, 5, and 6.

[3] http://www.cs.sunysb.edu/ algorith/implement/lpsolve/implement.shtml

Algorithm 2 Cutting Plane Inference($M, \mathbf{G}^0, \mathbf{x}$)

$i \leftarrow 0$
$\mathbf{y}' \leftarrow 0$
repeat
  $i \leftarrow i + 1$
  $\mathbf{y} \leftarrow solve(\mathbf{G}^{i-1}, \mathbf{x})$
  if $s(\mathbf{x}, \mathbf{y}) > s(\mathbf{x}, \mathbf{y}')$ **then**
    $\mathbf{y}' \leftarrow \mathbf{y}$
  end if
  for $each(\phi, w) \in M$ **do**
    $\mathbf{G}^i \leftarrow \mathbf{G}^{i-1} \cup Separate(\phi, w, \mathbf{x}, \mathbf{y})$
  end for
until $\mathbf{G}^i = \mathbf{G}^{i-1}$ or $i > maxIterations$
return $\mathbf{y}'$

3.3.5 Discriminative Weight Learning of Markov Logic

In MLNs, the method of learning weight discriminatively is proposed in (Singla and Domingos, 2005). In many applications, we know *a priori*; which predicates will be evidence and which ones will be queried. The goal of training is to correctly predict the latter given the former. If we define evidence atoms as $\mathbf{X}$ and query atoms as $\mathbf{Y}$, the *conditional likelihood* of $\mathbf{Y}$ given $\mathbf{X}$ is

$$P(\mathbf{Y} = \mathbf{y}|\mathbf{X} = \mathbf{x}) = \frac{1}{Z_{\mathbf{x}}} \exp\left(\sum_i w_i n_i(\mathbf{x}, \mathbf{y})\right) = \frac{1}{Z_{\mathbf{x}}} \exp\left(\sum_j w_j g_j(\mathbf{x}, \mathbf{y})\right) \tag{3.34}$$

where ϕ_Y is the set of all MLN clauses with at least one grounding involving a query atom, $n_i(\mathbf{x}, \mathbf{y})$ is the number of true groundings of the ith clause involving query atoms. G_Y is the set of ground clauses in $M_{L,C}$, and $g_i(\mathbf{x}, \mathbf{y}) = 1$ if the jth ground clause is true and 0 otherwise. The gradient of the conditional log-likelihood (CLL) is

$$\begin{aligned} \frac{\partial}{\partial w_i} \log P(\mathbf{Y} = \mathbf{y}|\mathbf{X} = \mathbf{x}) &= n_i(\mathbf{x}, \mathbf{y}) - \sum_{\mathbf{y}'} P_w(\mathbf{Y} = \mathbf{y}'|\mathbf{X} = \mathbf{x}) n_i(\mathbf{x}, \mathbf{y}') \\ &= n_i(\mathbf{x}, \mathbf{y}) - E_w[n_i(\mathbf{x}, \mathbf{y})] \end{aligned} \tag{3.35}$$

Computing the expected counts $E_w[n_i(\mathbf{x}, \mathbf{y})]$ is intractable but they can be approximated by the counts $n_i(\mathbf{x}, \mathbf{y}^*_w)$ in the MAP state $y^*_w(\mathbf{x})$ (i.e., the most probable state of $\mathbf{y}$ given

$\mathbf{x}$). If most of the probability mass of $P_w(\mathbf{y}|\mathbf{x})$ is concentrated around $\mathbf{y}^*_w(\mathbf{x})$. Hence, computing the CLL now requires only MAP inference to find $\mathbf{y}^*_w(\mathbf{x})$. If the training data is broken up into separate examples, an MAP inference per example is performed. This approach was initially proposed by Collins (2002) and called structured perceptron algorithm. In MLNs, MaxWalkSAT or ILP with CPI deals with the MAP inference, as we mentioned in Section 3.3.4.

Perceptron is one of the *online learning* algorithms. Online learning updates weight parameters $\mathbf{w}$ in instance-by-instance manner. For every instance i in the dataset, we perform the following two steps:

1. we perform MAP inference to find the best hidden solution $\hat{\mathbf{y}}$ given the current parameter $\mathbf{w}$ and the observation $\mathbf{x}$;

2. we compare solution $\hat{\mathbf{y}}$ with the *gold* solution $\mathbf{y}_t$ and update weights $\mathbf{w}$ based on this comparison.

A very simple way to update the weights $\mathbf{w}$ of a linear discriminative function $\mathbf{w} \cdot \Phi(\mathbf{x}, \mathbf{y})$ is perceptron update rule (Collins, 2002)

$$\mathbf{w}_{i+1} = \mathbf{w}_i + \tau(\Phi(\mathbf{x}, \mathbf{y}_t) - \Phi(\mathbf{x}, \hat{\mathbf{y}})) \tag{3.36}$$

where τ is a parameter which controls the learning rate. If $\hat{\mathbf{y}} = \mathbf{y}_t$, then no update is performed. On the other hand, if $\hat{\mathbf{y}} \neq \mathbf{y}_t$ then the weight vector is moved further in the direction of the true feature vector $\mathbf{f}(\mathbf{x}, \mathbf{y}_t)$ and further away from the guessed feature $\mathbf{f}(\mathbf{x}, \hat{\mathbf{y}})$.

Instead of perceptron, we can exploit arbitrary algorithms of online learning. For weight learning of MLNs, Riedel (2008) used margin infused relaxation algorithm (MIRA) (Crammer and Singer, 2003) instead of structured perceptron algorithm. MIRA is a modified version of perceptron by considering large margin and MIRA generally performs better than perceptron algorithm. Below is a supplement explanation about MIRA.

MIRA updates weight vector w as follows:

$$\begin{aligned} &\min && \|\mathbf{w}_{i+1} - \mathbf{w}_i\| \\ &s.t. && \mathbf{w}_{i+1}\Phi(\mathbf{x}, \mathbf{y}_t) - \mathbf{w}_{i+1}\Phi(\mathbf{x}, \hat{\mathbf{y}}) \geq L(\mathbf{y}_t, \hat{\mathbf{y}}) \quad \forall \hat{\mathbf{y}} \neq \mathbf{y}_t \end{aligned} \tag{3.37}$$

where $\mathbf{w}_i$ is the current parameter vector and $\mathbf{w}_{i+1}$ is the updated parameter vector. That is, Equation 3.37 means that, in each iteration i, the score of the correct assignment must exceed the score of an incorrect assignment by (user-defined) loss function

$L(\mathbf{y}_i, \hat{\mathbf{y}})$ The loss function could, for example, be the number of wrong label in a temporal relation identification task. In comparison to many other training methods (such as optimizing the conditional likelihood), the advantage of online learning algorithms is relatively low computational cost with effective MAP inference methods.

Chapter 4

Japanese Event-Argument Relation Extraction

4.1 Introduction

Event-argument relation extraction is one of the challenging problems in Natural Language Processing. The analysis extracts semantic information such as "who did what to whom", which is often useful to various applications like information extraction, document summarization, and machine translation. We focus on verbal and adverbal events called *predicates* (not logical predicates) and analyze predicate-argument (PA) structures.

Predicate-argument relation extraction is often called semantic role labeling. In English, it has been researched on large corpora such as FrameNet (Fillmore et al., 2001) and PropBank (Palmer et al., 2005). Japanese PA relation extraction is a kind of semantic role labeling but an argument is often called *case*. A typical example of Japanese PA relation is shown in Figure 4.1.

In this example, "行った (went)" is a predicate and there are two arguments for the predicate, that is, a nominative case role (ga) is "彼 (He)" and a dative case role (ni) is "図書館 (library)".

In Japanese, PA annotated corpora such as Kyoto Text Corpus (Kawahara et al., 2002) and NAIST Text Corpus (Iida et al., 2007) have been developed and utilized.[1] Taira et al. (2008) and Imamura et al. (2009) tackled PA relation extraction on NAIST Text Corpus. They created three separated models corresponding to each of the case;

[1] Kyoto Text Corpus is annotated with surface cases and NAIST Text Corpus is annotated with deep cases

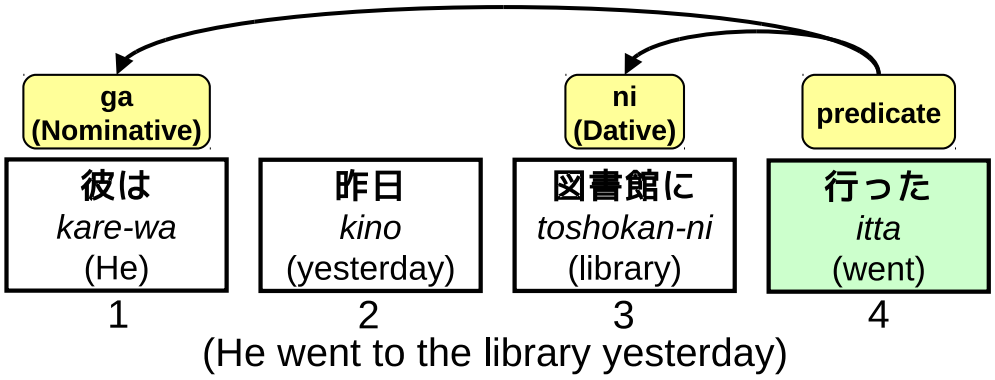


Figure 4.1: Example of Japanese Predicate-Argument Structure

ga (Nominative), wo (Accusative), and ni (Dative). Especially, Imamura et al. achieved high performance by exploiting selectional preference features extracted from large-scale unlabelled corpora.

Instead of exploiting large-scale corpora, we utilize important dependencies between one PA relation and another in the same sentence. In order to use such dependencies as global constraints, we apply a Markov Logic approach to Japanese PA relation extraction. In recent years, in English semantic role labeling, a Markov Logic model has achieved one of the state-of-the-art results (Meza-Ruiz and Riedel, 2009a). With global constraints between multiple PA relations, a Markov Logic model can avoid inconsistencies between several PA relations and improve performance of extraction.

In addition, we introduce new global constraints to effectively delete inappropriate argument candidates which are unrelated to PA relations. We consider that extraction of PA relations and deletion of the other phrases are two sides of the same coin. We jointly perform such extraction and deletion with Markov Logic.

Through our experiments, we report the effectiveness of the Markov Logic approach to Japanese PA relation extraction in detail. We show that our model with global constraints outperforms the model without them. Comparison with previous work shows that our Markov Logic approach achieves competitive results without selectional preference features obtained from large-scale unlabelled data. In qualitative analysis, we find that our global model resolves some difficult cases such as PA relations in relative clauses.

We summarize our main contributions in this chapter:

1. We expand a Markov Logic approach for Japanese PA relation extraction adding new global constraints to avoid considering phrases unrelated to PA relations

2. Through our quantitative and qualitative evaluation, we demonstrate that our global approach resolves some difficult cases and achieves competitive results compared with the state-of-the-art.

4.2 Background

The data we used in this work is from NAIST Text Corpus (NTC) (Iida et al., 2007). NTC is based on the same text as Kyoto Text Corpus (Kawahara et al., 2002), which contains 38,384 sentences from 2,929 news articles.[2] The annotation in NTC has the three case roles: "ga (Nominative)", "wo (Accusative)", and "ni (Dative)". The predicate-argument annotation in NTC is based on deep cases and is more difficult to analyze than the surface case annotations which Kyoto Text Corpus employs. Note that Kyoto Text Corpus includes morphological information, base phrase segmentation, and syntactic dependency structure. We can merge these annotation from Kyoto Text Corpus and deep case annotation from NTC.

There are two main previous work with NTC. First, Taira et al. (2008) researched extraction of PA relations by SVM classifiers and decision lists. Their approach focused on not only verbal predicates but also nominal predicates. Secondly, Imamura et al. (2009) combined a Maximum Entropy model and a language model learned from large-scale corpora and achieved the state-of-the-art results.

Both Taira et al. and Imamura et al. created an independent model for each of the cases *ga, wo*, and *ni* (the left box in Figure 4.2). So, their models neglect the dependencies between cases. For example, the method in previous work produces "NP2" for both ga and ni cases. Though it is unlikely that the same noun phrase occupies two argument positions of a predicate, it is possible with their models.

However, our Markov Logic approach creates a joint model for the three cases and finds the most probable assignments taking into consideration the dependency between them. As a result, our model can prevent such an unlikely result (See the right box in Figure 4.2).

Moreover, in contrast to Imamura's work, our method does not exploit large-scale corpora. They depended on their language model derived from large-scale corpora to decide the selectional preference between a predicate and an argument. On the other hand, we handle the problem by global optimization in a sentence without using large-scale corpora.

[2]These articles are from a Japanese newspaper, "Mainichi Shinbun"

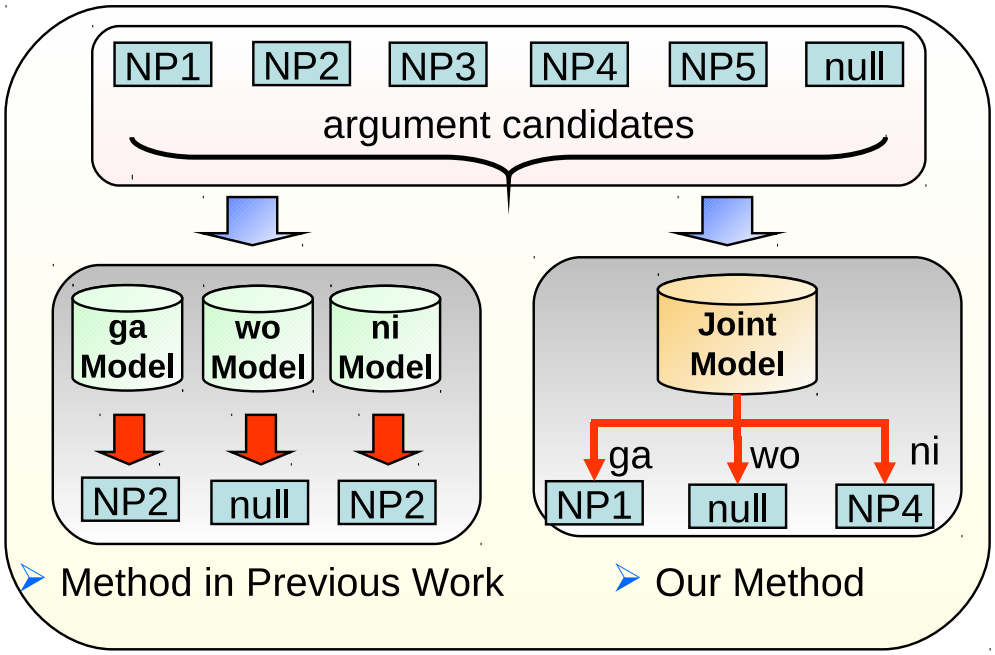


Figure 4.2: Difference between Previous Work and Our Method to Japanese PA Relation Extraction

On Kyoto Text Corpus, Kawahara and Kurohashi (2006b) proposed a probabilistic model for surface case structure analysis. In addition, there are some researches of automatic case-frame acquisition incoorprating large-scale data (Sasano et al., 2004; Kawahara and Kurohashi, 2004; Kawahara and Kurohashi, 2006a; Sasano et al., 2009; Kawahara and Kurohashi, 2010). However, in this thesis, we focus on deep-case strucure analysis on NAIST Text Corpus without unlabeled data.

In the CoNLL Shared Task 2009 (Hajič et al., 2009), a competition of multilingual semantic role labeling was held and Japanese was one of the target languages. In the shared task, Meza-Ruiz and Riedel (2009b) proposed a joint approach with Markov Logic. They also reported their Markov Logic approach for English semantic role labeling in detail (Meza-Ruiz and Riedel, 2009a). Their method divided the problem into four subtasks: predicate identification, argument identification, sense disambiguation, and role labeling. The subtasks are solved jointly.[3] We adapt their model to Japanese PA relation extraction. In order to compare with Taira et al. (2008) and Imamura et al. (2009), we perform only argument identification and role labeling. Note that the

[3]Note, in the CoNLL *2009* Shared Task, predicate identification is not necessary. So, they used the CoNLL *2008* Shared Task data in this work.

Japanese annotation of CoNLL Shared Task 2009 was also based on surface case and easier than that in NTC. Our Markov Logic model is novel as a Japanese PA relation extractor for deep case annotation.

4.3 Proposed Method

This section describes our proposed method. We propose Markov Logic model for Japanese PA relation extraction. Different from the other tasks in Chapters 5 and 6, we will not propose SVM pipeline model because previous work (Taira et al., 2008) already proposed it.

We will describe our proposed Markov Logic Network (MLN) in detail. First, let us define logical predicates for our MLN. Note, in order to distinguish predicates – verbal and adverbial events– from logical predicates, we sometimes call them *linguistic* predicates. The three *Hidden* predicates are listed in Table 4.1.

Table 4.1: Hidden Predicates for Japanese Predicate-Argument Relation Extraction

logical predicate	definition
$isArg(i)$	Bunsetsu i is an argument
$delete(i)$	Bunsetsu i is deleted
$role(i, j, r)$	Bunsetsu i has an argument j with role r

Note that Japanese dependency parsing is based on *bunsetsu* units, which are similar in concept to English base phrases. In order to exploit information parsed in this way, we handle all logical predicates by bunsetsu phrases (not words).

The hidden predicates model the decisions we need to make: whether a bunsetsu phrase i is an argument of some (linguistic) predicates (argument identification); whether a bunsetsu phrase i is deleted (phrase deletion); whether a bunsetsu phrase j is an argument of the predicate i with semantic role r (role labeling).

Here the first two types of decision can be modeled through unary logical predicates $isArg(a)$ and $delete(i)$, while the other type can be represented by a ternary logical predicate $role(p, a, r)$. Because we do not know their information at test time, we call them *hidden*.

Our Markov Logic approach is based on English semantic role labeling with Markov Logic as proposed by Meza-Ruiz and Riedel (2009a). As mentioned earlier, they divided the problem into four subtasks and defined five hidden predicates (*isPredicate,*

Table 4.2: Observed Predicates for Japanese PA Relation Extraction

logical predicate	description	example
$word(i, w)$	Bunsetsu i has word form w	行った (went), 彼 (he)
$stem(i, s)$	Bunsetsu i has stem s	行く (go)
$pos(i, p)$	Bunsetsu i has POS tag p (coarse-grained)	名詞 (noun)
$dpos(i, p)$	Bunsetsu i has POS tag p (fine-grained)	名詞-一般 (noun-general)
$ne(i, n)$	Bunsetsu i has named entity tag n (from NE tagger)	PERSON，LOCATION
$kana(i, k)$	Bunsetsu i has kana (Romanization) k	itta, kare
$isPred(i)$	Bunsetsu i is a predicate	True or False
$numeric(i)$	Bunsetsu i has a number character	True or False
$definite(i)$	Bunsetsu i contains the article corresponding to DEFINITE "the", such as "sore" or "sono"	True or False
$demonstrative(i)$	Bunsetsu i contains the article corresponding to DEMONSTRATIVE "this" or "that", such as "kono" or "ano"	True or False
$particle(i)$	Bunsetsu has a particle such as "wa","ga","wo", "ni"	True or False
$goiCate(i, g)$	Bunsetsu i has lexical category tag g in Nihongo Goi Taikei (Ikehara et al., 1997)	スポーツ (sport), 女性 (female)
$goiMatch(i, j, r)$	Bunsetsu phrases i and j satisfy the selectional restriction for r in Nihongo Goi Taikei	True or False
$dep(i, j, d)$	Dependency label between i and j is d	True or False
$path(i, j, l)$	Syntactic path between i and j is l	↑↓ (sibling), ↑↑ (ancestor)

isArgument, hasRole, role, and *sense*). In order to be comparable with the previous work in Japanese PA relation extraction (Taira et al., 2008; Imamura et al., 2009), we deal with only argument identification and role labeling in our research. Therefore, we define only the three hidden predicates in Table 4.1.

In addition to the hidden predicates, we define *observed* logical predicates representing information at test time. For example, in our case we could introduce a logical predicate $word(i, w)$ which indicates that a phrase i has the word form w. We list the all observed predicates in Table 4.2.

With our logical predicates defined, we can now go on to incorporate our intuition

Table 4.3: Global Formulae of *isArg* and *role* for Japanese PA Relation Extraction

Formula	Description
$isArg(a) \Rightarrow \exists p.\exists r.role(p,a,r)$	Every argument must relate to at least one predicate
$role(p,a,r) \Rightarrow isArg(a)$	If a plays the role r for p, then a has to be an argument
$role(p,a,r_1) \wedge r_1 \neq r_2 \Rightarrow \neg role(p,a,r_2)$	There is exactly one case role between a predicate and an argument

about the task using weighted first-order logic formulae. In the following we will explain the formulae of our proposed MLN. Sections 4.3.1 and 4.3.2 describe our local and global formulae, respectively. Section 4.3.3 mentions the formulae for deletion.

4.3.1 Local Formulae

We say that a formula is *local* if its groundings relate any number of observed ground predicates to exactly one hidden ground predicate. Local formulae are defined with some observed predicates from Table 4.2 and a hidden predicate from Table 4.1.

The local formulae for *isArg* and *delete* capture the relation of the bunsetsu phrases with their lexical and syntactic properties (simple phrase property). The formula describing a local property of word form is

$$word(a,+w) \Rightarrow isArg(a) \tag{4.1}$$

which implies that a bunsetsu a is an argument with a weight that depends on the word form. Note, the + notation indicates that the MLN contains one instance of the rule, with a separate weight, for each assignment of the variables with a plus sign.

The local formulae for *role* represent properties between two bunsetsu phrases (linked phrases property). For example, the following formula

$$ne(a,+n) \wedge dep(p,a,+d) \Rightarrow role(p,a,+r) \tag{4.2}$$

denotes a local property of named entity and syntactic dependency.

As in Formula (4.2), some observed predicates (*goiMatch, dep*, and *path*) in Table 4.2 construct formulae using other observed predicates in this table.

First-order logical formulae such as Formulae (4.1) and (4.2) become the feature templates of MLN. Each template produces several instantiations. An example of a template instantiation based on Figure 4.1 is

$$ne(1, PERSON) \wedge dep(4, 1, \text{“}D\text{”}) \Rightarrow role(4, 1, \text{“ga”}) \tag{4.3}$$

which is a typical expansion from Formula (4.2).

Moreover, as shown in Table 4.2, *goiMatch* implements a *selectional restriction* feature. Nihongo Goi Taikei (Ikehara et al., 1997) is a Japanese Thesaurus which covers constraints between the predicates and arguments. *goiMatch* is utilized as follows:

$$goiMatch(p, a, r) \Leftarrow role(p, a, r). \tag{4.4}$$

Selectional restriction is a constraint which examines the validity of an argument for a predicate. For example, we have the follwing sentence,

John	X	ate
ジョンは	X を	食べた
1	2	3

(John ate a X.)

where a variable X is arbitrary words (nouns). Here we want to evaluate the selectional restriction between “食べた (ate)” and “X を (X)”. Such restriction is

$$goiMatch(3, 2, \text{“}wo\text{”}) \Leftarrow role(3, 2, \text{“}wo\text{”}) \tag{4.5}$$

which examine the selectional restriction of Wo-case (Accusative).

If X takes “車 (car)”, Formula 4.5 becomes false because “車を食べた (ate a car)” is curious and selectional restriction in Nihongo Goi Taikei is not satisfied. On the other hand, if X takes “リンゴ (apple)”, Formula 4.5 can be true.

4.3.2 Global Formulae

The intuition behind the previous formulae can also be captured using a local classifier.[4] However, Markov Logic also allows us to say more:

$$isArg(a) \Rightarrow \exists p.\exists r.role(p, a, r) \tag{4.6}$$

[4]Consider a log-linear binary classifier with a “*PERSON*” feature: here for every phrase i the decision “i is an argument” becomes more likely with a higher weight for this feature.

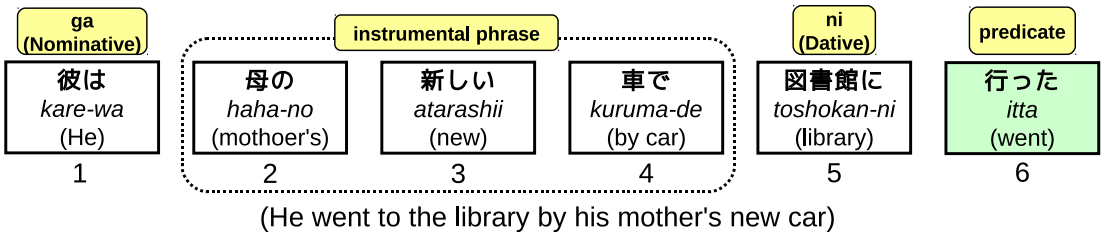


Figure 4.3: Example of Japanese PA Relation with Instrumental Case

In this formula, we made a statement about more global properties of a PA relation extraction that cannot be captured with local classifiers. This formula ensures the consistency between (linguistic) predicate and argument, that is, arguments belong to at least one predicate. This type of rule forms the core idea of our global model.

Global formulae involve two or more atoms of hidden predicates and enable us to jointly deal with argument identification, phrase deletion, and role labeling. With global formulae, our MLN considers not only a single decision at a time but also handles several decisions, simultaneously. Our global formulae for argument identification and role labeling are shown in Table 4.3.

The formulae in Table 4.3 are hard constraints which enforce consistency between the hidden predicates. In MLN, formulae of hard constraint are defined as special formulae with *infinite* weights. A possible world which violates hard constraints is never chosen as a correct answer. For example, Formula (4.6) is such a global formula. Another formula ensuring the consistency between *role* and *isArg* is

$$role(p, a, r) \Rightarrow isArg\,(a) \tag{4.7}$$

which indicates "If a phrase a plays the role r for p, then a must be an argument".

The last global formula

$$role(p, a, r_1) \wedge r_1 \neq r_2 \Rightarrow \neg role(p, a, r_2) \tag{4.8}$$

implies that there is only one case role between a linguistic predicate p and an argument a. Formula (4.8) enables us to prevent the contradiction shown in Figure 4.2.

4.3.3 Deletion Formulae

The main idea of our deletion is to delete bunsetsu phrases which are unrelated to PA relations and to help extract correct arguments. Extraction of correct arguments

and deletion of non-arguments are two sides of the same idea. An example is shown in Figure 4.3. We have a main verb "行った (went)" as a linguistic predicate and there are five argument candidates for it. We want to extract correct arguments, "彼は (He)" for ga-case and "図書館 (library)" for ni-case among the five candidates. Here, if we can remove an instrumental case, "母の新しい車で (by mother's new car)", extracting the correct arguments becomes much easier.

Notably, our significant contribution is doing this deletion processes with extraction of PA relations, simultaneously. Deleting too many bunsetsu phrases often hurts the recall because it often deletes correct arguments. We call this phenomena *over-deletion*. Performing extraction and deletion by one joint model prevents over-deletion and improves the performance of PA relation extraction.

Local Deletion Formulae

Deletion formulae are also divided into local and global. However, local formulae implement the same properties for *isArg* we mentioned in Section 4.3.1. As an exception, a characteristic local formula is

$$dep(i, j, +d) \wedge isPred(j) \Rightarrow \neg delete(i). \tag{4.9}$$

which implies the PA relations with syntactic dependencies are not deleted. It implements the fact that PA relations often have syntactic dependency relations. Actually, we can find that dependency relations are dominant in Table 4.5 and Formula (4.9) contributes to improve performance.

However, the local formulae address the deletion of a single bunsetsu phrase and we cannot expect a large improvement by adding *delete*. The main contributions of *delete* come from the global deletion formulae.

Global Deletion Formulae

The global formulae for *delete* have the three hard and one soft constraints. We show the global formulae in Table 4.4. The first three formulae in this table show the hard constraints which ensure the consistency between *delete* and the other two hidden predicates (*isArg* and *role*). The most important formula of them is

$$delete(i) \Rightarrow \neg isArg(i) \tag{4.10}$$

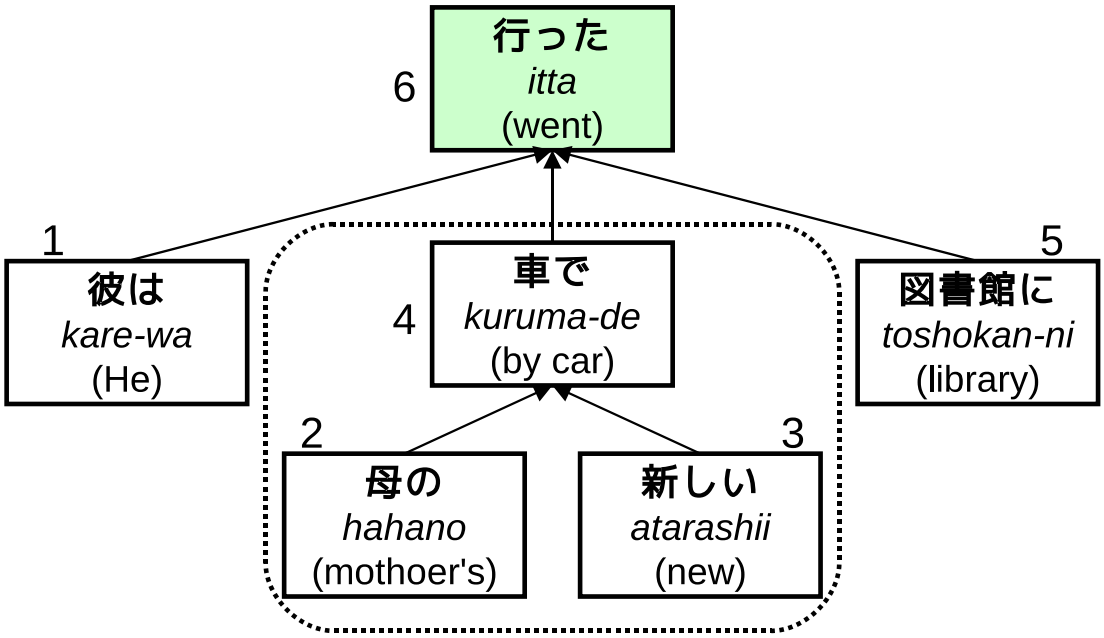


Figure 4.4: Deletion of Instrumental Case in Japanese PA

Table 4.4: Global Deletion Formulae for Japanese PA Relation Extraction

Formula	Description
$isArg(a) \Rightarrow \neg delete(a)$	If a is an argument then it is not deleted.
$delete(i) \Rightarrow \neg isArg(i)$	If a bunsetsu i is deleted then it is not an argument.
$role(p,a,r) \Rightarrow \neg delete(p) \wedge \neg delete(a)$	If a is an argument of p with the role r then neither p nor a is deleted.
$word(h,+w) \wedge pos(h,+p) \wedge dep(h,m,+d) \wedge delete(h) \Rightarrow delete(m)$	If a head phrase h is deleted with word w and POS p then a child phrase m is deleted.

which implies that the deleted phrase does not become an argument.

The last formula in Table 4.4 is defined as a soft constraint:

$$word(h,+w) \wedge pos(h,+p) \wedge dep(h,m,+d) \wedge delete(h) \Rightarrow delete(m) \quad (4.11)$$

which denotes "if a head phrase h is removed, then the child phrases m should be deleted". This formula does not always hold but the remaining uncertainty with regard

to this formula is captured by a weight trained from corpora. This constraint implements the important deletion concept as we mentioned earlier.

Considering the example in Figure 4.3, Formula (4.11) is grounded as,

$$\begin{aligned} &word(4, \text{“車で”}) \wedge pos(4, \text{NOUN+PARTICLE}) \\ &\wedge dep(4, 2, \text{“}D\text{”}) \wedge delete(4) \Rightarrow delete(2) \end{aligned} \tag{4.12}$$

which implies that “if ‘車で (by car)’ is removed, ‘母の (mother’s)’ should be also removed”. Figure 4.4 shows the dependency parsed tree extracted from the sentence in Figure 4.3. The subtree under “車で (by car)” should be deleted by Formula (4.12).

Note that Japanese dependency parsing usually targets only unlabeled parsing. Almost all labels are “D”.[5] Therefore, we exploit the *word* and *pos* of head bunsetsu phrases as a substitution. In Japanese, word form and POS implicitly give us information similar to dependency labels. However, if we exploit our method in English, labeled information such as *probj* or *amod* should be helpful to train proper weights for Formula (4.11).

4.4 Experimental Setup

Our experimental setting is based on previous work (Taira et al., 2008; Imamura et al., 2009) which was performed on NAIST Text Corpus.

Taira et.al. exploited local classifiers with Support Vector Machines (SVMs). Imamura et.al. applied a Log-Linear Model (Maximum Entropy method) as classifiers and used a large scale unlabeled data. In general, predicate-argument structure analysis includes identifying bunsetsus of predicates (predicate identification). However, both Taira et.al. and Imamura et.al., performed PA relation extraction given bunsetsus of predicates. Accordingly, we also follow their setting and do not perform predicate identification.

Let us summarize our used data and tools. The data used, NAIST Text Corpus version 1.4β, has news articles and editorials. As training examples, we use articles published from January 1st to January 11th and editorials from January to August. As development data, we use articles published on January 12th and 13th and editorials in September. For evaluation, we use articles dated January 14th to 17th and editorials dated October to December. This way to split the data is same as Taira et al. (2008). We show the statistics of the evaluation data in Table 4.5.

[5] We sometimes have “P”, “A”, and “I” labels but it is not enough to model our deletion idea.

Table 4.5: Statistics in Evaluation Data (Test Set of NAIST Text Corpus)

	ga	wo	ni
Dep.	13,086	5,192	3,645
Zero-Intra	4,556	376	231
Total	17,642	5,568	3,876

As seen in this table, "ga-case" is dominant. PA relations which have syntactic dependency relations (Dep.) are much more common than zero-anaphoric PA relations (Zero-Intra). Note that we target only PA relations which occur in a sentence (*intra-sentential* PA relations). The joint approach using Markov Logic is computationally hard even if it targets only *intra-sentential* PA relations. Therefore, extraction of *inter-sentential* PA relations which are crossing sentence boundaries is intractable. Moreover, our approach finds the most optimized PA assignments in a whole sentence. To keep consistency in a sentence, we delete the sentences which have inter-sentential PA relations.

For extracting features, we exploit the annotation of Kyoto Text Corpus as the POS and the syntactic dependency of bunsetsu phrases. We perform named entity tagging using CaboCha.[6] Based on Taira's work, we introduce selectional restriction features from a Japanese Thesaurus, Nihongo Goi Taikei (Ikehara et al., 1997). Learning and inference algorithms for our joint model are provided by Markov thebeast, a Markov Logic engine tailored for NLP applications.

4.5 Experimental Results

4.5.1 Impact of Global Formulae

First, let us show the comparison between the models with/without global constraints in Table 4.6. ***Global*** is the model with global constraints and ***Local*** is without them. Note that the local and global formulae of deletion are also included in ***Local*** and ***Global***, respectively. Table 4.6 shows Precision (P), Recall (R), and F1-value (F) of each hidden predicate. We can find that ***Global*** yielded clear improvements for all hidden predicates. These improvements are statistically significant.[7] These re-

[6] `http://chasen.org/~taku/software/cabocha/`

[7] $\rho < 0.01$, McNemar's test 2-tailed

Table 4.6: Local vs Global of Japanese PA Relation Extraction

	Local			***Global***		
	P	R	F	P	R	F
isArg	79.2	71.4	75.1	94.6	84.2	89.1
delete	86.6	90.4	88.4	94.3	97.9	96.1
role	86.3	72.5	78.8	85.5	**77.7**	81.4

Table 4.7: Effect of Hidden Predicate Removal in Japanese PA Relation Extraction

Predicate Removed	P	R	F
No removal (***Global***)	85.5	77.7	81.4
-isArg	**84.8**	77.9	81.2
-delete	85.3	**76.8**	80.8
-isArg-delete (***Local***)	86.3	72.5	78.8

Table 4.8: Runtime of Japanese Event Argument Relation Extraction (sec.)

	Local	***Global***
Train	5792.5	8684.2
Test	1009.3	1165.1

sults suggest that the three target subtasks (argument identification, phrase deletion, and role labeling) can cooperate with each other. For PA relation extraction (*role*), the recall was mainly improved (the value in bold type).

We perform a simple analysis of hidden predicate removal. For each hidden predicate, a model was trained with that predicate removed and all other predicates retained. For PA relation extraction (*role*), Table 4.7 shows the model performance with removal of the *isArg* and *delete* predicates.

The removal of *delete* drops the model performance larger than that of *isArg*. While the removal of *isArg* drops the precision and saves the recall, the removal of *delete* works the other way around.

Let us show the runtimes of our models. Table 4.8 shows the runtimes for our ***Local*** and ***Global*** models. We take the averages over three times running for training and testing. In general, the inferences with global constraints are more complex than those without them and such complex inferences increase runtimes of training and testing. Actually, our ***Global*** model takes more time than ***Local*** model both in training and

Table 4.9: Comparison to the State-of-the-Art for Japanese PA Relation Extraction (F1)

	Local			***Global***			[Taira, 2008]			[Imamura, 2009]		
	ga	wo	ni	ga	wo	ni	ga	wo	ni	ga	wo	ni
Dep.	85.7	91.2	79.5	**88.8**	91.3	79.7	75.6	88.2	**89.5**	87.0	**93.9**	80.8
Zero-Intra	42.1	7.3	0.0	**54.1**	10.3	0.0	30.2	11.4	**3.7**	50.0	**30.8**	0.0

testing. For the runtime comparison between Markov Logic approach and the other approaches, we will show some results in Chapter 5 and 6.

4.5.2 Comparison to the State-of-the-art

Next, we evaluate the results of PA relation extraction (*role*) by each case, "ga (Nominative)", "wo (Accusative)", and "ni (Dative)" in Table 4.9. All scores in the table are F1-value. Our ***Global*** model is more advantageous in "Zero-Intra" than ***Local*** model. Especially, in ga-case of Zero-Intra the score jumped from 42.1pt to 54.1pt (+12pt). Again, with global constraints, our global model finds the most probable state in the sentence. It is often difficult to extract Zero-Intra PA relations with only local features because syntactic dependencies between them are weak. Therefore, our global constraints contribute to finding correct assignments of PA relations and we got a large improvement in Zero-Intra.

Let us compare our results with the state-of-the-art (Taira et al., 2008; Imamura et al., 2009). In Table 4.9, we show the best scores in bold types for each case. For ga-case, our model, ***Global***, outperformed the others. On the other hand, for wo-case and ni-case, our results were relatively lower than them. Because our approach deals with the all three cases by one joint model and ga-case is dominant in the data, it extracts more numbers of ga-case than the others. However, ga-case is often the most important for PA relation extraction and sometimes called *indispensable case*. Our method can extract such important information better than previous work. Although our model did not exploit large-scale corpora, our results are competitive to the results of Imamura et al. (2009).

4.6 Discussion

In this section, we will mainly discuss the qualitative aspect of our results.

(this) (reason) (Gray Wolf) (**revival in the US**) (plan) (**FWS**) (in Canada)
この ため ， 灰色狼の 米復活を 進める 魚類野生動物局が カナダで
1 2 3 4 5 6 7

(capture) (wild) (**twelve wolves**) (transport by air)
捕獲した 野性の 十二匹を 空輸 .
8 9 10 11

(Form this reason, FWS which plans to revive Gray Wolf in the US captured twelve wolves in Canada and transported them by air.)

In the above sentence, we have three predicates (gray boxed) and three arguments (underlined). The relations between predicates and arguments are complex with relative clause and often cause misunderstandings.

About this sentence, our ***Local*** model output:

$$\Big\{role(5,6,ga), role(5,4,wo), role(8,6,ga), \underline{role(11,2,ga)}, role(11,10,wo)\Big\}$$

It did not output wo-case of "捕獲した (capture)". Because we do not have case-frame dictionary in NTC, our models did not know that "捕獲した" usually requires wo-case (Accusative).

Another error is underlined that ga-case of "空輸 (transport by air)" is identified as "ため (reason)", because "ため" is only a phrase dependent on "空輸".

On the other hand, ***Global*** improved the errors as

$$\Big\{role(5,6,ga), role(5,4,wo), role(8,6,ga), \underline{role(8,10,wo)}, \underline{role(11,6,ga)}, role(11,10,wo)\Big\}.$$

By global optimization in a sentence, our ***Global*** model overcame the lack of semantic features and successfully identified "十二匹を" as wo-case of "捕獲した". This PA relation is in a relative clause and often hard to identify. Though Abekawa and Okumura (2005) resolved Japanese PA relations in relative clauses by exploiting large-scale corpora, our Markov Logic approach handles this problem by global optimization. Moreover, in global model, $\{delete(1), delete(2), delete(7)\}$ are also output and "この" and "ため" did not become argument candidates. As a result, "魚類野性動物局が" was correctly selected as a ga-case of "空輸".

4.7 Summary

In this chapter, we proposed a new Markov Logic approach for Japanese predicate-argument (PA) relation extraction. Our model exploited global constraints between multiple PA relations and introduced phrase deletion. Our global constraints successfully improved the performance of PA relation extraction. In comparison to the state-of-the-art, our approach achieved competitive results with no large-scale data.

As a future direction, incorporating large-scare unlabelled data should be effective. Selectional preference features from large-scale corpora are expected to improve the performance for extracting wo-case and ni-case.

In related to our deletion approach, the state-of-the-art techniques of sentence compression are worth to be considered. It might be interesting to evaluate our approach in sentence compression tasks. Adding sentence compression might make the PA relation extractor more efficient and allow us to extract *inter-sentential* PA relations, too.

Chapter 5

Biomedical Event Extraction

5.1 Introduction

The increasing amount of biomedical texts results from high throughput experiments. This situation demands the automatic extraction of useful information from these texts by Natural Language Processing techniques. One of the more recent information extraction tasks is biomedical event extraction. With the introduction of the GENIA Event Corpus (Kim et al., 2008) and the BioNLP'09 shared task data (Kim et al., 2009), a set of documents annotated with events and their arguments, various approaches for event extraction have been proposed (Björne et al., 2009; Buyko et al., 2009; Poon and Vanderwende, 2010).

Previous work has considered the problem on a per-sentence basis and neglected information from other sentences in the same document. These information are possibly important to understand the document. In particular, no previous work has considered using coreference information to improve event extraction. Here we propose a new approach to extract event-argument (E-A) relations that uses coreference information.

Our approach is built on two main ideas:

1. extracting coreferent arguments based on *salience in discourse*
2. predicting arguments over sentence boundaries with the help of a *substitutability* relation.

First, noun phrases (NPs) that corefer with other NPs have an implicit significance in discourse structures based on Centering Theory (Grosz et al., 1995). Significant

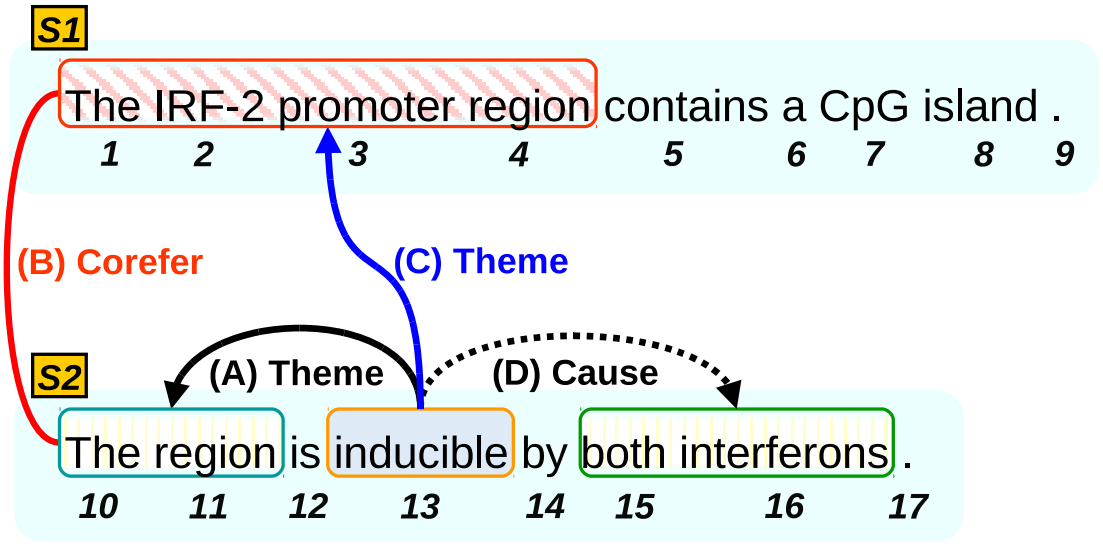


Figure 5.1: Cross-Sentence Event-Argument Relation Extraction in a Biomedical Document

entities are highly likely to be mentioned multiple times. We call this kind of significance "*salience in discourse*." *Salience in discourse* is a useful criterion for measuring the importance of entity mentions, and this criterion gives our E-A relation extractors a higher chance to extract arguments which are coreferent with other mentions. When considering discourse structure, arguments which are coreferent to something (e.g. "The region" in Figure 5.1) also have higher *salience in discourse*. They are hence more likely to be arguments of other events mentioned in the document. Using this information helps us to identify the correct arguments for candidate events and increases the likelihood of extracting arguments with antecedents corresponding to the Arrow (A) in Figure 5.1. Note that identifying coreferent arguments is not just important to improve the F1 score of event-argument relation extraction: assuming that *salience in discourse* indicates the novel information the author wants to convey, these are the arguments we should extract at any cost.

Secondly, *substitutability* is a property of event-argument relations such that the relation between an event and its argument is substitutability across coreference relations. It enables us to extract cross-sentence mentions as arguments of events. Previous work on this task has primarily focused on identifying event-arguments within a sentence. However cross-sentence event-argument relations are common, for example see Figure 5.1. It illustrates an example of E-A relation extraction including cross-sentence E-A.

In the sentence *S2*, we have "inducible" as an event to be identified. When identifying intra-sentence arguments in *S2*, we obtain "The region" as Theme and "both interferons" as Cause. However, in this example, "The region" is not optimal as a Theme because "The region" is coreferent to "The IRF-2 promoter region" in *S1*. Thus, the true Theme of "inducible" is "The IRF-2 promoter region" as this phrase is more informative as an argument. In this case, "The region" is just an anaphor of the true argument. The idea of *substitutability* entails that if "The region" is a Theme of "inducible" and "The region" is coreferent to "The IRF-2...", then "The IRF-2..." is also a Theme of "inducible". It allows us to extract cross-sentence E-A relations such as the Arrow (C) in Figure 5.1.

We propose two models which implement these ideas to extract event-argument (E-A) relations involving coreference information. One is based on local classification with SVM, and another is based on a joint Markov Logic Network (MLN). To remain efficient, and akin to existing approaches, both look for events on a per-sentence basis. However, in contrast to previous work, our models consider as candidate arguments not only the tokens of the current sentence, but also all tokens in the previous sentences that are identified as antecedents of some tokens in the current sentence.

5.2 Background

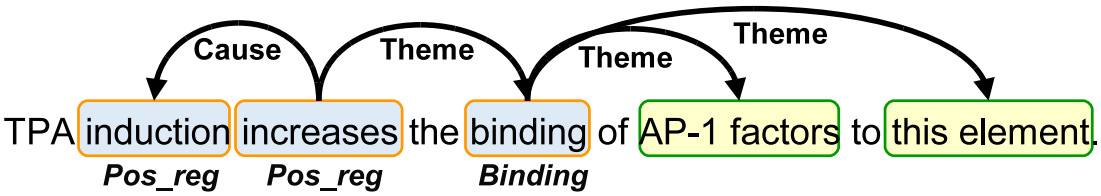


Figure 5.2: An Example of Biomedical Event Extraction

5.2.1 Task Definition

Event extraction on biomedical text involves three sub-tasks; identification of event trigger words; classification of event types; extraction of the arguments of the identified events (E-A). Figure 5.2 shows an example of event extraction. In this example, we

have three event triggers: “induction”, “increases”, and “binding”. The corresponding event types are *Positive_regulation (Pos_reg)* for “induction” and “increases”, and *Binding* for “binding”. In Figure 5.2, “increases” has two arguments; “induction” and “binding”. The roles we have to identify fall into two classes: “Theme” and “Cause”. In the case of our example the roles of the two arguments of “increases” are Cause and Theme, respectively.

Note that a large number of nominal events can be found in biomedical corpora. For example, in Figure 5.2 the arguments of “increases” are both nominal events. Such events which are arguments of other events are often hard to identify.

5.2.2 Biomedical Corpora for Event Extraction

There are two major corpora for biomedical event extraction: The GENIA Event Corpus (GEC) (Kim et al., 2008), and the data of the BioNLP’09 shared task.[1] The latter is in fact derived from the GEC. There are some important differences between them.

event type GEC has fine-grained event type annotations (35 classes), while BioNLP’09 data focuses on only 9 event subclasses.

non-event argument BioNLP’09 data does not differentiate between protein, gene and RNA, while the GEC corpus does.

coreference annotation Both GEC and BioNLP’09 corpora provide coreference annotations related to event extraction. However, in the case of the BioNLP’09 data coreference information primarily concerns protein names and abbreviations that follow in parenthesis. The GEC, on the other hand, provides proper cross-sentence coreference. Moreover, the sheer number of coreference annotations is much larger. Björne et al. (Björne et al., 2009) also mentioned that coreference relations could be helpful for cross-sentence E-A extraction but the coreference annotation necesary to train a coreference resolver is not present in BioNLP’09 data.

For our work we choose the GEC, primarily because of the amount and quality of coreference information it provides. This allows us to train a coreference resolver, as well as testing our hypothesis when gold coreference annotations are available. The

[1] `http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/SharedTask/`

second reason to prefer GEC over the BioNLP'09 corpus is its fine-grained annotation. We believe that this setting is more realistic.

5.2.3 Issues of Previous Work

Various approaches have been proposed for event-argument relation extraction on biomedical text. However, even the current state-of-the-art does not exploit coreference relations and focuses exclusively on intra-sentence E-A extraction.

BioNLP'09 has three tasks 1, 2, and 3. Task 1 is core event extraction and mandatory. Our work also focuses on Task 1. For example, Björne et al. achieved the best results for Task 1 in the BioNLP'09 competition (Björne et al., 2009). However, they neglected all cross-sentence E-A. They also reported that they did try to detect cross-sentence arguments directly without the use of coreference. This approach did not lead to a reasonable performance increase.

In BioNLP'09, Riedel et al. proposed a joint Markov Logic Network to tackle the task (Riedel et al., 2009). Their system makes use of global features and constraints, and performs event trigger and argument detection jointly. Poon and Vanderwende (Poon and Vanderwende, 2010) also applied Markov Logic and achieved competitive performance to the state-of-the-art result of Björne (Björne et al., 2009). However, in both cases no cross-sentence information is exploited.

To summarize, so far there has been no research within biomedical event extraction which exploits coreference relations and tackles cross-sentence E-A relation extraction. By contrast, for predicate-argument relation extraction in a Japanese newswire text corpus,[2] Taira et al. do consider cross-sentence E-A extraction (Taira et al., 2008). However, they directly extract cross-sentence links without considering coreference relations. Moreover, their approach is based on a pipeline of SVM classifiers, and their performance on cross-sentence E-A extraction was generally low (Low 20s% F1).

5.2.4 The Direction of Our Work

We present a new approach that exploits coreference information for E-A relation extraction. Moreover, in contrast to previous work on the BioNLP'09 shared task we apply our models in a more realistic setting. Instead of relying on gold protein

[2] http://cl.naist.jp/nldata/corpus/

annotations, we use a Named Entity tagger; and instead of focusing on the coarse-grained annotation of the BioNLP task, we work with the GEC corpus and its fine-grained ontology.

From now on, for brevity, we refer to cross-sentence event-argument relations as "*cross-links*" and intra-sentence event-argument relations as "*intra-links*".

We propose two coreference-based models. One is an SVM based model that extracts intra-links first and then cross-links as a post-processing step. The other is a joint model defined with Markov Logic that jointly extracts intra-links and cross-links and allows us to model salience of discourse in a principled manner.

5.3 Methods

We have two ideas for incorporating coreference information into E-A relation extraction,

- Extracting valuable E-A relations based on "*salience in discourse*"
- Predicting cross-links by using "*substitutability*" including coreference relations

Salience in discourse is the idea of considering how important the occurring mentions are. We exploit it as a likelihood of arguments of events. *substitutability* is a property of event-argument relations such that the relation between an event and its argument is substitutable across coreference relations. It enables us to identify the E-A relations over sentence boundaries. According to these ideas, we propose two approaches. One is a pipeline model based on SVM classifiers, and the other is a joint model based Markov Logic Network.

5.3.1 SVM Pipeline Model

In our pipeline model we apply the SVM model proposed by (Björne et al., 2009). Their original model first extracts events and event types with a multi-class SVM (1st phase). Then it identifies the relations between all event-protein and event-event pairs by another multi-class SVM (2nd phase). Note that in our setting, the 1st phase classifies event types into 36 classes (35 types + "Not-Event"). Moreover, while protein annotations were given in the BioNLP'09 shared task, for the GEC we extract them

using an NE tagger. The features we used for the 1st and 2nd phases are summarized in the first and the second columns of Table 5.2, respectively.

After identifying intra-links, the pipeline model deterministically attaches, for each intra-sentence argument of an event, all antecedents inside/outside the current sentence. We implement *substitutability* as a post-processing step. However, it is difficult for the SVM pipeline to implement the idea of *salience in discourse*. We believe that a Markov Logic model is preferable in this case. We utilize linear kernel as kernel function of SVMs.

5.3.2 MLN Joint Model

Because of the time and space complexities, it is difficult to construct Markov Logic Networks for joint E-A relation extraction and coreference resolution across a complete document. Hence we follow two strategies: (1) restriction of argument candidates based on coreference relations; (2) construction of a joint model which collectively identifies intra-links and cross-links. Restricting argument candidates helps us to construct a very compact yet still effective model. A joint model enables us to simultaneously extract intra-links and cross-links and contributes to the performance improvement. In addition, we will see that this setup still allows us to implement the idea of *salience in discourse* with global formulae in Markov Logic.

Predicate Definition Our joint model is based on the model proposed by Riedel et al. (2009). We first define the predicates of the proposed Markov Logic Network (MLN). There are three *hidden* predicates corresponding to what the target information we want to extract (Table 5.1).

Table 5.1: Hidden Predicates of Biomedical Event Extraction

role(i, j, r)	token i has an argument j with role r
event(i)	token i is an event
eventType(i, t)	token i is an event with type t

In this work, *role* is the primary hidden predicate since it represents event-argument relations.

Next we define *observed* predicates representing information that is available at both train and test time. We define $coref(i, j)$, which indicates that token i is coreferent to

Table 5.2: Used Local Features for SVM Pipeline and MLN Joint of Biomedical Event Extraction

Description	SVM 1st phase *event* & *eventType*	SVM 2nd phase *role* (E-A)	MLN predicate
Word Form	X	X	$word(i, w)$
Part-of-Speech	X	X	$pos(i, p)$
Word Stem	X	X	$stem(i, s)$
Named Entity Tag	X	X	$ne(i, n)$
Chunk Tag	X	X	$chunk(i, c)$
In Event Dictionary	X	X	$dict(i, d)$
Has Capital Letter	X	X	$capital(i)$
Has Numeric Characters	X	X	$numeric(i)$
Has Punctuation Characters	X	X	$punc(i)$
Character Bigram	X		$bigram(i, bi)$
Character Trigram	X		$trigram(i, tri)$
Dependency label	X	X	$dep(i, j, d)$
Labeled dependency path between tokens		X	$path(i, j, pt)$
Unlabeled dependency path between tokens		X	$pathNL(i, j, pt)$
Least common ancester of dependency path		X	$lca(i, j, L)$

token j (they are in the same entity cluster). $coref(i, j)$ obviously plays an important role in our coreference-based joint model. We list the remaining observed predicates in the last column of Table 5.2.

Our MLN is composed of several weighted formulae that we divide into two classes. The first class contains local formulae for *event, eventType*, and *role*. We say that a formula is local if it considers only one single hidden ground atoms. The formulae in the second class are global: they involve two or more atoms of hidden predicates. In our case they consider *event, eventType*, and *role* atoms simultaneously.

Basic Local Formulae Our local features are based on features employed in previous work (Björne et al., 2009; Riedel et al., 2009) and listed in Table 5.2. We exploit two types of formula representation: "simple token property" and "link tokens property" defined by (Riedel et al., 2009).

The first type of local formulae describes properties of only one token and such properties are represented by the predicates in the first section of Table 5.2. The second

Table 5.3: Basic Global Formulae of Biomedical Event Extraction

Formula	Description
$\text{event}(i) \Rightarrow \exists t.\text{eventType}(i,t)$	If there is an event there should be an event type
$\text{eventType}(i,t) \Rightarrow \text{event}(i)$	If there is an event type there should be an event
$\text{role}(i,j,r) \Rightarrow \text{event}(i)$	If j plays the role r for i then i has to be an event
$\text{event}(i) \Rightarrow \exists j.\text{role}(i,j,Theme)$	Every event relates to need at least one argument.

Table 5.4: Coreference Formulae of Biomedical Event Extraction

Symbol	Name	Formula	Description
(*SiD*)	*Salience in Discourse*	$corefer(j,k) \Rightarrow \exists i.\text{role}(i,j,r) \wedge \text{event}(i)$	If a token j is coreferent to another token k, there is at least one event related to token j
(*Sub*)	*Substitutability*	$\text{role}(i,j,r) \wedge corefer(j,k) \Rightarrow \text{role}(i,k,r)$	If j plays the role r for i and j is coreferent to k then k also plays the role r for i
(*FC*)	*Feature Copy*	$corefer(j,k) \wedge F(k,+f) \Rightarrow \text{role}(i,j,r)$	If j is coreferent to k and k has feature f then j plays the role r for i

type of local formulae represents properties of token pairs and linked tokens property predicates (dep, $path$, $pathNL$, and lca) in the second section of Table 5.2.

Basic Global Formulae Our global formulae are designed to enforce consistency between the three *hidden* predicates and are shown in Table 5.3. Riedel et al. (Riedel et al., 2009) presented more global formulae for their model. However, some of these do not work well for our task setting on the GENIA Event Corpus. We obtain the best results by only using global formulae for ensuring consistency of the hidden predicates.

5.3.3 Involving Coreference Information

We explain our coreference-based approaches using the example in Figure 5.1. First, the two intra-links in *S2* are represented by $\text{role}(13, 11, \text{Theme})$ – Arrow (A) and $\text{role}(13, 15, \text{Cause})$ – Arrow (D). Note, in these terms, phrasal arguments are driven by *anchor* tokens which are the ROOT tokes on dependency subtrees of the phrases. The coreference relation is represented by $\text{corefer}(11, 4)$ – Bold Line (B). Finally, the cross-link is represented by $\text{role}(13, 4, \text{Theme})$ – Arrow (C).

With the example in Figure 5.1, we explain the two main concepts : *Salience in Discourse (*SiD*)* and *substitutability (*Sub*)*. We also present an additional idea, *Feature Copy (*FC*)*.

Salience in Discourse The entities mentioned over and over again are important in discourse and accordingly highly likely to be arguments of some events. In order to implement this idea of *salience in discourse*, we add the Formula (*SiD*), shown in the first row of Table 5.4. Formula (*SiD*) requires that if a token j is coreferent to another token k, there is at least one event related to token j. Our model with Formula (*SiD*) prefers coreferent arguments and aggressively connects them with events. Note that our coreference resolver always extracts coreference relations which are related to events, since coreference annotations in GEC are always related to events.

Substitutability Another main concept is "*substitutability*", which is important for intra/cross-link extraction.

As mentioned earlier, the SVM pipeline enforces *substitutability* as a post-processing step.

For the MLN joint model, let us consider the example of Figure 5.1 again.

$$\text{role}(13, 11, \text{Theme}) \wedge \text{corefer}(11, 4) \Rightarrow \text{role}(13, 4, \text{Theme})$$

This formula denotes that, if an event "inducible" has "The region" as a Theme and "The region" is coreferent to "The IRF-2 promoter region", then "The IRF-2 promoter region" is also a Theme of "inducible". The three atoms, $\text{role}(13, 11, \text{Theme})$, $\text{corefer}(11, 4)$, and $\text{role}(13, 4, \text{Theme})$ in this formula correspond respectively to the three Arrows (A), (B), and (C) in Figure 5.1. This formula is generalized as Formula (*Sub*) shown in the second row of Table 5.4.

The merit of using Formula (*Sub*) is that we can take care of cross-links by only solving intra-links and using the associated coreference relations. The only candidate arguments of cross-links are the arguments which are coreferent to intra-sentence mentions (antecedents).

The improvement due to Formula (*Sub*) depends on the accuracy of the intra-link $\text{role}(i, j, r)$ and coreference relation $\text{corefer}(j, k)$ atoms. Clearly, this accuracy depends partially on the effectiveness of Formula (*SiD*) above. It should also be clear that the improvement due to Formula (*SiD*) is also affected by Formula (*Sub*) because *Sub* impacts the condition $\exists i.\text{role}(i, j, r)$ in Formula (*SiD*). Thus, the formulae representing *Salience in Discourse* and *substitutability* interact with each other.

Feature Copy We make additional use of coreference information through "*Feature Copy*". The main idea is to supplement the features of an anaphor by adding the features of its antecedent. According to the example of Figure 5.1, the formula,

$$\text{corefer}(11, 4) \wedge \text{word}(4, \text{“IRF-2”}) \Rightarrow \text{role}(13, 11, \text{Theme})$$

describes a word feature "IRF-2" to the anaphor "The region" in *S2*. Here word(i, w) represents a feature that the child token of the token i on the dependency subtree is word w. To be exact, this formula allows us to employ additional features of the antecedent to solve the link role(13, 11, Theme). This formula is generalized as Formula (*FC*) in the last row of Table 5.4. In Formula (*FC*), F denotes the predicates which represent basic features such as word, POS, and NE tags of the tokens. Formula (*FC*) copies the features of cross-sentence arguments (antecedents) to intra-sentence arguments (anaphors). *Feature Copy* is not a novel idea but it helps improve performance. For the SVM pipeline model we add equivalent features.

5.3.4 Coreference Resolution

In our work, we introduce a simple coreference resolver based on a pairwise coreference model (Soon et al., 2001). It employs a binary classifier which classifies all possible pairs of noun phrases into "corefer" or "do not corefer". Popular external resources like WordNet often do not work well in the biomedical domain. Hence, our resolver identifies coreference relations using only basic features such as word form, POS, and NE tag. We use SVM-struct for learning and testing the binary classifiers. In this model, negative examples often overwhelm positive ones, and we therefore select a value over 10000 for the C-parameter. We achieve 59.1 pairwise F1 on GENIA Event Corpus evaluating 5-fold cross validation.

There is some previous work on coreference resolution for biomedical domains (Yang et al., 2004; Su et al., 2008). They constructed original coreference annotations for learning and testing. Their models use much richer features for machine learning classifiers and their systems achieve better results with around 70 F1. However, owing to the differences of the data used, it is difficult to directly compare their results with ours. Moreover, using the richer feature they propose, we would likely see improvements in our system as well. Finally, we confirm that there is enough room for improvement by also evaluating with gold coreference annotations.

Note that we optimize our resolver for event extraction because our event extractors require high precision results from coreference resolution. For the SVM model, coreference resolution errors directly hurt performance. For MLN model, noisy results from coreference resolution often disturb the coreference formulae when learning weights. We noticed that the weights of coreference formulae remain small when the coreference resolution results have less than 70 precision and our MLN event extractor rarely obtains cross-sentence event-argument relations as a result. Some features and string distance metrics may enable us to better balance precision and recall, but we attach greater importance to precision. As a result, our high precision resolver achieves over 90 for precision but lower than 50 for recall.

5.4 Experimental Setup

Let us summarise the data and tools we employ. The data for our experiments is the *GENIA Event Corpus (GEC)* (Kim et al., 2008). For feature generation, we employ the following tools. POS and NE tagging are performed with the *GENIA Tagger*,[3] for dependency path features we apply the *Charniak-Johnson reranking parser with a Self-Training parsing model*,[4] This model is optimized for biomedical parsing and achieves 84.3pt F1 on GENIA corpus (McClosky and Charniak, 2008). We convert the parsed results to dependency tree using the *pennconverter tool*.[5] Learning and inference algorithms for joint model are provided by *Markov thebeast* (Riedel, 2008), a Markov Logic engine tailored for NLP applications. Our pipeline model employs *SVM-struct*[6] both in learning and testing. As we mentioned in the previous section, for coreference resolution, we also employ SVM-struct for binary classification.

Figure 5.3 shows the structure of our experimental setup. Our experiments perform the following steps. (1) First we perform preprocessing (tagging and parsing). (2) Then we perform coreference resolution for all the documents and generate lists of token pairs that are coreferent to each other. (3) Finally, we train the event extractors: SVM pipeline (SVM) and MLN joint (MLN) involving coreference relations. We evaluate all systems using 5-fold cross validation on GEC.

[3] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/tagger/
[4] http://www.cs.brown.edu/~dmcc/biomedical.html
[5] http://nlp.cs.lth.se/software/treebank_converter/
[6] http://www.cs.cornell.edu/People/tj/svm_light/svm_struct.html

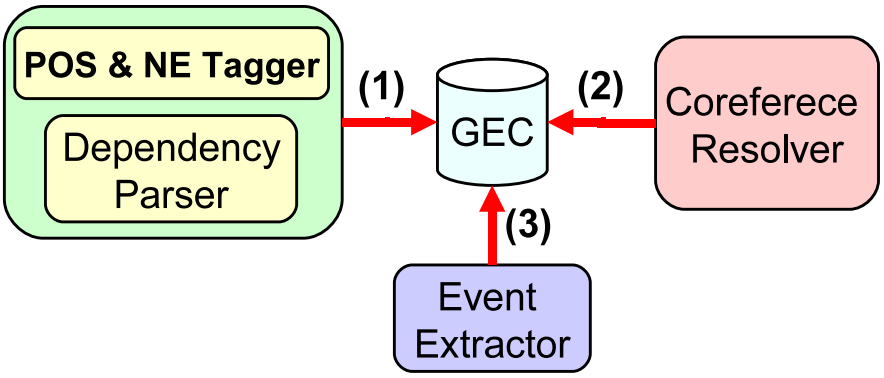


Figure 5.3: Experimental Setup of Biomedical Event Extractor

Table 5.5: Results of Biomedical Event Extraction (F1)

System	Coreference	event	eventType	role
(a) SVM	NONE	77.0	67.8	52.3 (0.0)
(b) SVM	SYS	77.0	67.8	53.6 (**+1.3**)
(b′) SVM	GOLD	77.0	67.8	55.4 (+3.1)
(c) MLN	NONE	80.5	70.6	51.7 (0.0)
(g) MLN	SYS	80.8	70.8	53.8 (**+2.1**)
(g′) MLN	GOLD	81.2	70.8	56.7 (+5.0)

5.5 Experimental Results

In the following we will first show the results of our models for event extraction with/without coreference information. We will then present more detailed results concerning E-A relation extraction.

5.5.1 Impact of Coreference Based Approach

We begin by showing the SVM and MLN results for event extraction in Table 5.5. We present F1-values of event, eventType, and role (E-A relation) extraction. The three columns (event, eventType, and role) in Table 5.5 correspond to the *hidden* predicates in Table 5.1.

Let us consider rows (a)-(b) and (c)-(g). They compare the SVM and MLN approaches with and without the use of coreference information. The column “Corefer”

Table 5.6: Three Types of Biomedical Event-Argument (EA) Structure

Type	Description	Edge in Figure 5.1
Cross	E-A relations crossing sentence boundaries (cross-link)	Arrow (C)
W-ANT	Intra-sententence E-As (intra-link) with antecedents	Arrow (A)
Normal	Neither Cross nor W-ANT	Arrow (D)

indicates how the coreference information is used: "NONE"– without coreference; "SYS"– with coreference resolver; "GOLD"– with gold coreference annotations.

We note that adding coreference information leads to 1.3 point F1 improvement for the SVM pipeline, and 2.1 point improvement for MLN joint. Both improvements are statistically significant ($\rho < 0.01$, McNemar's test 2-tailed). With gold coreference information, systems (b′) and (g′) clearly achieve more significant improvements.

Let us move on to the comparisons between SVM pipeline and MLN joint models. For event and eventType we compare row (b) with row (g) and observe that the MLN outperforms the SVM. This is to be contrasted with results for the BioNLP'09 shared task, where the SVM model (Björne et al., 2009) outperformed the MLN (Riedel et al., 2009). This contrast may stem from the fact that GEC events are more difficult to extract due to a large number of event types and lack of gold protein annotations, and hence local models are more likely to make mistakes that global consistency constraints can rule out.

For role extractions (E-A relation), SVM pipeline and MLN joint show comparable results, at least when not using coreference relations. However, when coreference information is taken into account, the MLN profits more. In fact, with gold coreference annotations, the MLN outperforms the SVM pipeline by a 1.3 point margin.

5.5.2 Detailed Results for Event-Argument Relation Extraction

Table 5.6 shows the three types of E-A relations we evaluate in detail.

They correspond to the arrows (A), (C), and (D) in Figure 5.1, respectively. We show the detailed results of E-A relation extraction in Table 5.7. All scores shown in the table are F1-values.

Table 5.7: Results of Biomedical E-A Relation Extraction (F1)

System	Corefer	Cross	W-ANT	Normal
(a) SVM	NONE	0.0	56.0	53.6
(b) SVM	SYS	**27.9**	57.0	54.3
(b′) SVM	GOLD	**54.1**	57.3	55.4
(c) MLN	NONE	0.0	49.8 (0.0)	53.2
(d) MLN	*FC*	0.0	51.5 (+1.7)	53.7
(e) MLN	*FC+SiD*	0.0	54.6 (+4.8)	53.3
(f) MLN	*FC+Sub*	36.7	51.7 (+1.9)	53.7
(g) MLN	*FC+SiD+Sub*	**39.3**	56.5 (**+6.7**)	54.3
(g′) MLN	GOLD	**69.7**	66.7 (**+16.9**)	55.3

SVM pipeline Model

The first part of Table 5.7 shows the results of the SVM pipeline with/without coreference relations. Systems (a), (b) and (b′) correspond to the first three rows in Table 5.5, respectively. We note that the SVM pipeline manages to extract cross-links with an F1 score of 27.9 points with coreference information from the resolver. The third low in Table 5.7 shows the results of the system with gold coreference which is extended from System (b). With gold coreference, the SVM pipeline achieves 54.1 points for "Cross". However, the improvement we get for "W-ANT" relations is small since the SVM pipeline model employs only *Feature Copy* and *Substitutability* concepts. In particular, it cannot directly exploit *Salience in Discourse* as a feature.

MLN joint Model

How does coreference help our MLN approach? To answer this question, the second part of Table 5.7 shows the results of the following six systems. The row (c) corresponds to the fourth row of Table 5.5 and shows results for the system that does not exploit any coreference information. Systems (d)-(g) include Formula (*FC*). In the sixth (e) and the seventh (f) rows, we show the scores of MLN joint with Formula (*SiD*) and Formula (*Sub*), respectively. Our full joint model with both (*SiD*) and (*Sub*) formulae comes in the eighth row (g). System (g′) is an extended system from System (g) with gold coreference information.

By comparing Systems (d)(e)(f) with System (c), we note that *Feature Copy (FC)*,

Salience in Discourse (SiD), and *Substitutability (Sub)* formulae all successfully exploit coreference information. For "W-ANT", Systems (d) and (e) outperform System (c), which establishes that both *Feature Copy* and *Salience in Discourse* are sensible additions to an MLN E-A extractor. On the other hand, for "Cross (cross-link)", System (f) extracts cross-sentence E-A relations, which demonstrates that *Substitutability* is important, too. Next, for cross-link, our full system (g) achieved 39.3 points F1 score and outperformed System (c) with 6.7 points margin for "W-ANT". The further improvements with gold coreference are shown by our full system (g′). It achieved 69.7 points for "Cross" and improved System (c) by 16.9 points margin for "W-ANT".

SVM Pipeline vs MLN Joint

The final evaluation compares SVM pipeline and MLN joint models. Let us consider Table 5.7 again. When comparing System (a) with System (c), we notice that the SVM pipeline (a) outperforms the MLN joint model in "W-ANT" without coreference information. However, when comparing Systems (b) and (g) (using coreference information by the resolver), MLN result is very competitive for "W-ANT" and 11.4 points better for "Cross".

Furthermore, with gold coreference, the MLN joint (System (g′) outperforms the SVM pipeline (System (b′)) both in "Cross" and "W-ANT" by a 15.6 points margin and a 9.4 points margin, respectively. This demonstrates that our MLN model will further improve extraction of cross-links and intra-links with antecedents if we have a better coreference resolver. Note that the MLN model has advantages over the SVM model especially when higher recall is required. We have 2,124 links of "Cross" and 2,748 of "W-ANT" for the evaluation of Table 5.7. MLN model-System (g′) finds 1,236 correct "Cross" and 1,778 correct "W-ANT" links. The SVM model-System (b′) finds only 833 correct links for "Cross" and 1,149 for "W-ANT".

Table 5.8 shows the runtime comparison between SVM pipeline and MLN joint models. SVM pipeline and MLN joint models for calculating the runtimes correspond to Systems (b) and (g) in Table 5.5, respectively. Again, we take the averages over three times running for training and testing. SVM pipeline model has two phases but the total runtimes of the 1st and 2nd phases are much shorter than those of MLN joint model. In biomedical domain, a sentence has more tokens than that in general domains. Global optimization in such a long sentence accordingly requires long time. However, the runtime depends on the problem we want to solve. We has another results

Table 5.8: Runtime Comparison between SVM Pipeline vs MLN Joint in Biomedical Event Extraction (sec.)

	SVM 1st phase	SVM 2nd phase	MLN
Train	1376.8	1277.4	7040.1
Test	21.2	22.1	1008.8

in Chapter 6.

5.6 Discussion

For lack of technical knowledge of biomedicine, it is difficult for us to analyze our results qualitatively. Instead of qualitative analysis, we briefly discuss some reasons of our results in this section.

The main topic is why our MLN joint model outperformed SVM pipeline? We consider that the reason for these results are two crucial differences between the SVM and MLN models:

- With Formula (*SiD*) in Table 5.4, MLN joint has more chances to extract "W-ANT" relations. It also effects the first term of Formula (*T*). By contrast, the SVM pipeline cannot easily model the notion of *salience in discourse* and the effect from coreference is weak.

- Formula (*Sub*) of MLN is defined as a soft constraint. Hence, other formulae may reject a suggested cross-link from Formula (*Sub*). The SVM pipeline deterministically identifies cross-links and is hence more prone to errors in the intra-sentence E-A extraction.

Finally, the potential for further improvement through a coreference-based approach is limited by the performance on intra-links extraction. Moreover, we also observe that the 20% of cross-links are cases of zero-anaphora. Here the utility of coreference information is naturally limited, and our Formula (*Sub*) cannot come into effect due to missing corefer(j,k) atoms.

5.7 Summary

In this chapter we presented a novel approach to biomedical event extraction with the help of coreference relations. Our approach incorporates coreference relations through the concepts of *salience in discourse* and *substitutability*. The coreferent arguments we focused on are generally valuable for document understanding in terms of discourse structure and they should be extracted at all cost. We proposed two models: SVM pipeline and MLN joint. Both improved the attachments of intra-sentence and cross-sentence related to coreference relations. Furthermore, we confirmed that improvements of coreference resolution lead to the higher performance of event-argument relation extraction.

However, potential for further improvement through a coreference-based approach is limited by the performance of intra-sentence links and zero-anaphora cases. To overcome these problems, we need to construct a collective approach that works on the full document. Specifically, we should construct a joint model of coreference resolution and event extraction considering all tokens in a document based on the idea of Narrative Schemas (Chambers and Jurafsky, 2009). If we take into account all tokens in a document at the same time, we can consider various relations between events (event chains) through anaphoric chains. But to implement such a joint model in Markov Logic, we will have to cope with the time and space complexities that arise in such a setting.

Chapter 6

Temporal Relation Identification

6.1 Introduction

This section describe a work on extracting event-event relation. We focus our effort on temporal relation as a typical example of event-event relations. The extraction of temporal relation is called temporal relation identification or temporal ordering. Temporal relation identification involves the prediction of temporal order between events and/or time expressions mentioned in text, as well as the relation between events in a document and the time at which the document was created.

Before we describe the details of temporal relation identification, we define the two main elements of this analysis, time expressions (TEs) and events. TEs are words or phrases which have obvious temporal information such as “today”, “July 4th”, and “two months ago.” Events are words or phrases which have indirect temporal information. Most of the events are expressed by verbs or verbals such as “happen”, “driving” and “the earthquake”.

Time and event expression analysis is regarded as an essential phase for understanding documents in semantic level, because temporal information often provides very important clues to identify semantic relations like causal relations. In the following example, there is strong interaction between temporal and causal relations.

> “In March 11th, she was injured her legs and went to a hospital, because a giant earthquake occurred and she was caught up in a building collapse.”

We map events and time expressions in the example sentence on Figure 6.1. People easily understand that “giant earthquake” and “building collapse” caused before “her

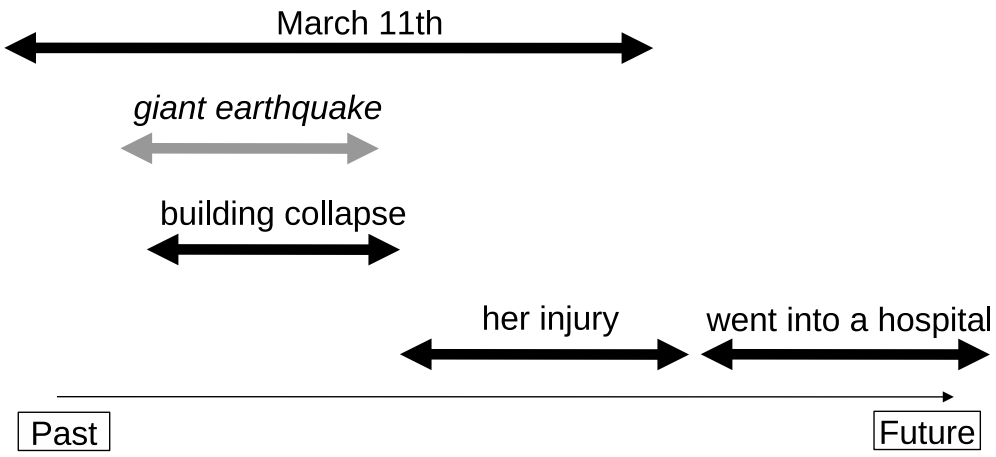


Figure 6.1: Events and time expressions mapping on a timeline

injury". And "her injury" results from "building collapse". But it is difficult for a machine learning system to identify such temporal ordering. The system, which does not know which events caused first, can not answer a simple question "What happened after the earthquake?" Thus, temporal information can be an essential point of document understanding.

In general, time normalization is a task coming after temporal relation identification. It maps all events and TEs to the real timeline. In the TERN-2004 Evaluation Workshop,[1] several time normalization techniques were proposed. However, the target of the workshop was only TEs. As an advanced task, it is necessary to map not only time but also events to the timeline.

As the first step of time and event expression analysis, in temporal relation identification task we identify the temporal order between two time and/or event expressions. This can be defined as a classification task with temporal relation labels based on Allen's time interval logic (Allen, 1983).

With the introduction of the TimeBank corpus (Pustejovsky et al., 2003a), a set of documents annotated with temporal information, it became possible to apply machine learning to temporal ordering (Boguraev and Ando, 2005; Mani et al., 2006). These tasks have been regarded as essential for complete document understanding and are useful for a wide range of NLP applications such as question answering and machine translation.

[1]TERN-2004 Evaluation Workshop – Time Expression Recognition and Normalization: `http://timex2.mitre.org/tern.html`

Most of these approaches follow a simple schema: they learn classifiers that predict the temporal order of a given event pair based on a set of the pair's of features. This approach is *local* in the sense that only a single temporal relation is considered at a time.

Learning to predict temporal relations in this isolated manner has at least two advantages over any approach that considers several temporal relations jointly. First, it allows us to use off-the-shelf machine learning software that, up until now, has been mostly focused on the case of local classifiers. Second, it is computationally very efficient both in terms of training and testing.

However, the local approach has an inherent drawback: it can lead to solutions that violate logical constraints we know to hold for any sets of temporal relations. For example, by classifying temporal relations in isolation we may predict that event X happened before, and event Y after, the time of document creation, but also that event X happened after event Y—a clear contradiction in terms of temporal logic. This logical constraint is illustrated in Figure 6.2.

In order to repair the contradictions that the local classifier predicts, Chambers and Jurafsky (2008b) proposed a global framework based on Integer Linear Programming (ILP). They showed that large improvements can be achieved by explicitly incorporating temporal constraints.

We proposed two approaches: one is a SVM pipeline model which solves a temporal relation exploiting the other relations as global features; the other is an MLN joint model which simultaneously solves several temporal relations with global constraints. The both approaches we propose in this thesis are similar in spirit to that of Chambers and Jurafsky: we seek to improve the accuracy of temporal relation identification by predicting relations in a more global manner. However, while they focused only on the temporal relations between events mentioned in a document, we also jointly predict the temporal order between events and time expressions, and between events and the document creation time.

The joint model also differs in another important aspect from the approach of Chambers and Jurafsky. Instead of combining the output of a set of local classifiers using ILP, we approach the problem of joint temporal relation identification using Markov Logic (Richardson and Domingos, 2006). In this framework global correlations can be readily captured through the addition of weighted first order logic formulae.

Using Markov Logic instead of an ILP-based approach has at least two advantages. First, it allows us to easily capture non-deterministic (soft) rules that tend to hold be-

tween temporal relations but do not have to. [2] For example, if event A happens before B, and B overlaps with C, then there is a good chance that A also happens before C, but this is not guaranteed.

Second, the amount of engineering required to build our system is similar to the efforts required for using an off-the-shelf classifier: we only need to define features (in terms of formulae) and provide input data in the correct format. [3] In particular, we do not need to manually construct ILPs for each document we encounter. Moreover, we can exploit and compare advanced methods of global inference and learning, as long as they are implemented in our Markov Logic interpreter of choice. Hence, in our future work we can focus entirely on temporal relations, as opposed to inference or learning techniques for machine learning.

We evaluate our approach using the data of the "TempEval" challenge held at the SemEval 2007 Workshop (Verhagen et al., 2007). This challenge involved three tasks corresponding to three types of temporal relations: between events and time expressions in a sentence (Task A), between events of a document and the document creation time (Task B), and between events in two consecutive sentences (Task C).

Our findings show that by incorporating global features or constraints that hold between temporal relations predicted in Tasks A, B and C, the accuracy for all three tasks can be improved significantly. In comparison to other participants of the "TempEval" challenge our approach is very competitive: for two out of the three tasks we achieve the best results reported so far, by a margin of at least 2%.[4] Only for Task B we were unable to reach the performance of a rule-based entry to the challenge. However, we do perform better than all pure machine learning-based entries.

6.2 Background

Temporal relation identification aims to predict the temporal order of events and/or time expressions in documents, as well as their relations to the document creation time (DCT). For example, consider the following (slightly simplified) sentence of Section 6.1.

[2] It is clearly possible to incorporate weighted constraints into ILPs, but how to learn the corresponding weights is not obvious.

[3] This is not to say that picking the right formulae in Markov Logic, or features for local classification, is always easy.

[4] To be slightly more precise: for Task C we achieve this margin only for "strict" scoring—see sections 6.4 and 6.5 for more details.

> With the introduction of the TimeBank corpus (Pustejovsky et al., 2003), machine learning approaches to temporal ordering became possible.

Here we have to predict that the "Machine learning becoming possible" event happened *AFTER* the "introduction of the TimeBank corpus" event, and that it has a temporal *OVERLAP* with the year 2003. Moreover, we need to determine that both events happened *BEFORE* the time this document was created.

Most previous work on temporal relation identification (Boguraev and Ando, 2005; Mani et al., 2006; Chambers and Jurafsky, 2008b) is based on the TimeBank corpus. The temporal relations in the Timebank corpus are divided into 11 classes; 10 of them are defined by the following 5 relations and their inverse: *BEFORE*, *IBEFORE (immediately before)*, *BEGINS*, *ENDS*, *INCLUDES*; the remaining one is *SIMULTANEOUS*. Such temporal annotations are based on Allen's time interval logic (Allen, 1983). Allen divided temporal relations between two time intervals into 13 classes. But Timebank omitted the two of them which are not shown in English texts.

In order to drive forward research on temporal relation identification, the SemEval 2007 shared task (Verhagen et al., 2007) (TempEval) included the following three tasks.

TASK A Temporal relations between events and time expressions that occur within the same sentence.

TASK B Temporal relations between the Document Creation Time (DCT) and events.

TASK C Temporal relations between the main events of adjacent sentences.[5]

To simplify matters, in the TempEval data, the classes of temporal relations were reduced from the original 11 to 6: *BEFORE, OVERLAP, AFTER, BEFORE-OR-OVERLAP, OVERLAP-OR-AFTER*, and *VAGUE*.

In this work we are focusing on the three tasks of TempEval, and our running hypothesis is that they should be tackled *jointly*. That is, instead of solving each task separately, we want to exploit the result of one task to the other tasks or to learn a single probabilistic model for all three tasks. This allows us to incorporate *transition rules* of temporal consistency that should hold across tasks. For example, if an event X happens *before* DCT, and another event Y *after* DCT, then surely X should have happened *before* Y. We illustrate this type of transition rule in Figure 6.2.

[5]The main event of a sentence is expressed by its syntactically dominant verb.

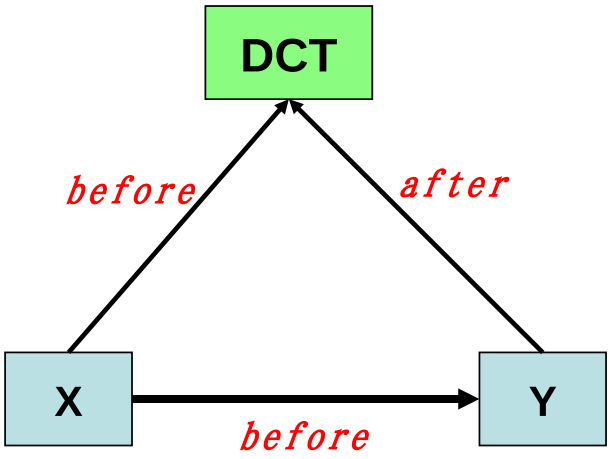


Figure 6.2: Example of Transition Rule for Temporal Relation Identification

Note that the correct temporal ordering of events and time expressions can be controversial. For instance, consider the example sentence again. Here one could argue that "the introduction of the TimeBank" may *OVERLAP* with "Machine learning becoming possible" because "introduction" can be understood as a process that is not finished with the release of the data but also includes later advertisements and announcements. This is reflected by the low inter-annotator agreement score of 72% on Tasks A and B, and 68% on Task C.

6.3 Methods

For the temporal relation identification tasks, we propose two approaches: SVM pipeline model with temporal relation paths; MLN joint model with global constraints of temporal closure. Both models exploit transition rules which are inference rules from the ordering property of time. An example of such rules is illustrated in Figure 6.2. The transition rule represented in Figure 6.2 is

$$(\text{X } \mathit{before} \text{ DCT}) \wedge (\text{Y } \mathit{after} \text{ DCT}) \Rightarrow (\text{X } \mathit{before} \text{ Y}) \tag{6.1}$$

which means if X happens *before* DCT and Y happends *after* DCT, then X happens *before* Y.

In order to represent the structures of transition rules, we define predicates (relation names) corresponding to edges of TempEval task chart. Let us describe the temporal

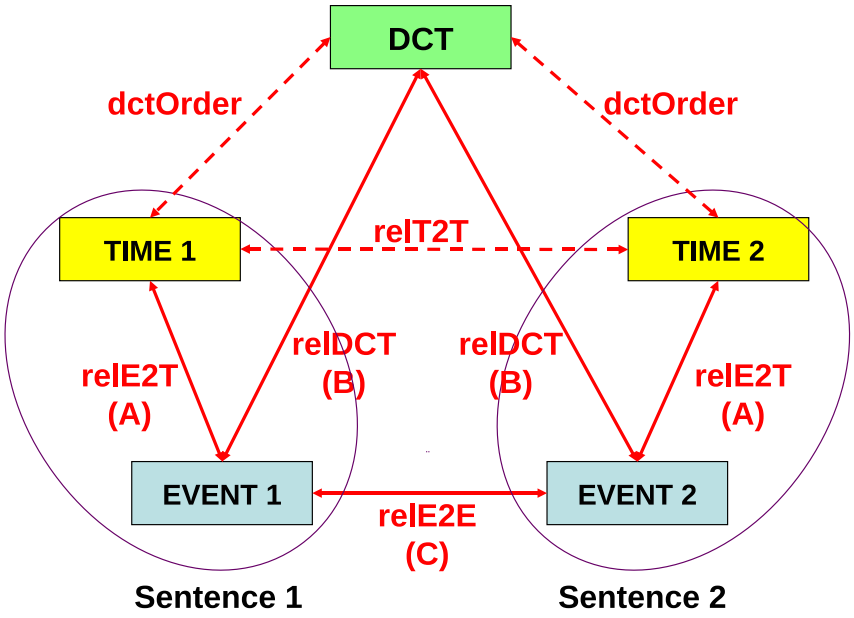


Figure 6.3: Relation Names with TempEval Tasks

relations in Figure 6.3. We describe the three relations (*hidden* predicates) corresponding to Tasks A, B, and C in Table 6.1.

Table 6.1: Hidden Predicates for Temporal Relation Identification

Task	Relation (predicate)	description
(A)	$relE2T(e,t,r)$	temporal relation r between an event e and a time expression t
(B)	$relDCT(e,r)$	temporal relation r between an event e and DCT
(C)	$relE2E(e_1,e_2,r)$	temporal relation r between two events of the same sentence, an event e_1 and another e_2
N/A	$relT2T(t_1,t_2,r)$	temporal relation r between two time expressions t_1 and t_2
N/A	$dctOrder(t,r)$	temporal relation r between a time expressions t and DCT

For Task (A), $relE2T(e,t,r)$ represents a temporal relation between an event e and temporal relation t is r. For Task (B), $relDCT(e,r)$ represents a temporal relation between an event e and a document creation time is r. For Task (C), $relE2E(e_1,e_2,r)$ represents a temporal relation between two events e_1 and e_2 is r.

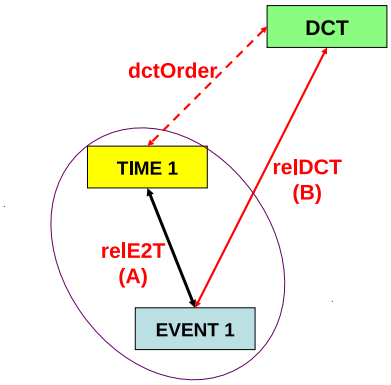


Figure 6.4: Temporal Relation Path for Task A

TempEval data contains two more types of temporal relations which are not supposed to be predicted: *relT2T* which represents a temporal relation between two time expressions; *dctOrder* which denotes a temporal relation between a time expression and a fixed DCT.

We explain the details of the two methods in the two following subsections.

6.3.1 SVM Pipeline Model

SVM pipeline model independently solves each three task but it exploits global *path* features. In this method, we presume that the complexity ordering of the three TempEval tasks is as follows: $B < A < C$. TempEval's final results (Verhagen et al., 2007) certainly show this order. In addition, we verified that by creating local classifiers for each task. Hence, we utilize the results of easier tasks for relatively difficult tasks: using the result of Task B for Task A and the results of Tasks A and B for Task C.

We propose *Temporal Relation Paths* which are composed of sequences of relation labels. Since the relation labels of Tasks A and B are easier to identify than those of Task C, the paths can become good clues to solve a more difficult task that is hard to solve only with static features.

More specifically, we explore the following relation paths (Figures 6.4, 6.5, 6.6, 6.7(a), and 6.7(b)).

Figure 6.4 shows the path for Task A ($dctOrder - (DCT) - relDCT$). The relDCT, a relation between EVENT1 and DCT, comes from results of Task B. We can acquire the dctOrder label by simply comparing both values of the TIMEX3 tags and deciding

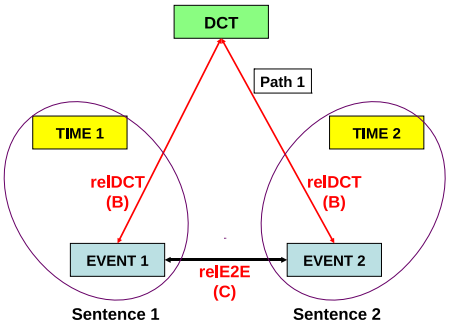


Figure 6.5: Path 1 for Task C

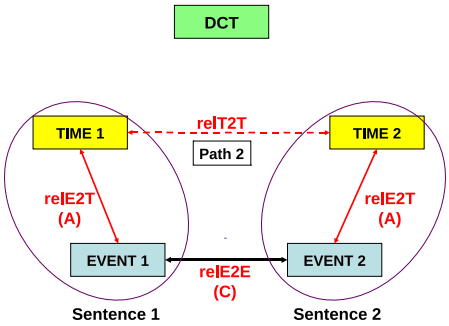
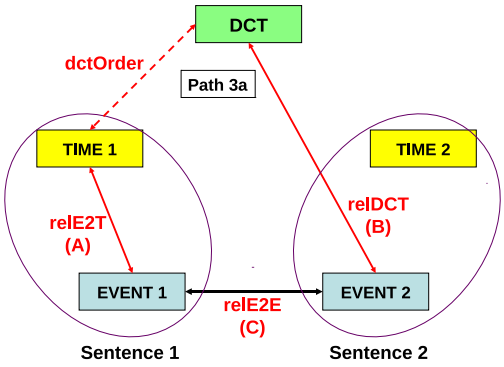
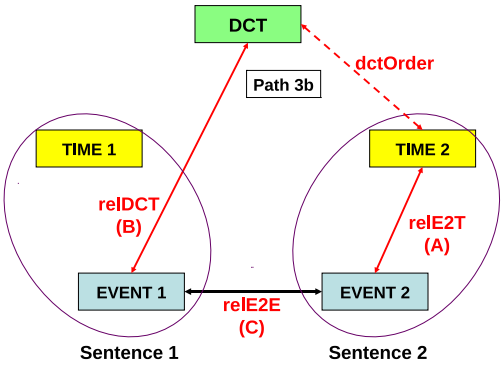


Figure 6.6: Path 2 for Task C



(a) Path 3a



(b) Path 3b

Figure 6.7: Temporal Relation Paths 3 for Task C

BEFORE, AFTER, or OVERLAP. As a result, this path is composed of three nodes including DCT (e.g., AFTER-(DCT)-AFTER).

Path 1 for Task C (Figure 6.5) uses the results of Task B directly and is composed of only the relations between DCT and the events ($relDCT - (DCT) - relDCT$; "DCT" is not a value but just a label). Introduction of Path 1 aims to use the "tense" information of the two target events. If the two target events are finite verbs and have the tense information, the accuracy of two relations on the Path 1 becomes precise. In such a case, the Path 1 will work very well.

In practice, the exact value of DCT is sometimes unavailable. However, Path 1 doesn't use the value of DCT. What we need is not the exact value of DCT but the relation between DCT and an event estimated by Task B. Paraphrasingly, we can use DCT as a pivot event without value.

Path 2 (Figure 6.6) uses the results of Task A and the relation between two time expressions ($relE2T - relT2T - relE2T$). Introduction of Path 2 is mainly for the

nominal events which do not have the tense information. We estimate the relation between the target event and the neighbouring time expression by Task A. The relation between TIME1 and TIME2 could be a very good clue for identifying the relation between the two main events that are anchoring on those time expressions. Path 2 is made of three label nodes (e.g., AFTER-OVERLAP-AFTER).

Paths 3a and 3b (Figure 6.7(a)(b)) use the both results of Tasks A and B (3a: $relE2T - dctOrder - (DCT) - relDCT$, 3b: $relDCT - (DCT) - dctOrder - relE2T$). Paths 3ab (Paths 3a and 3b) are intermediate between Path 1 and 2. Introduction of those paths are for identification of the relations between a finite verb and a nominal event. They are made of three label nodes (e.g., AFTER-OVERLAP-(DCT)-OVERLAP). These relation paths are the core of our first proposed method.

There are two reasons why we use the temporal relation paths as features instead of defining the strict transition rules and temporal closures as constraints. Temporal relation paths handle much more cases, including such cases that the strict transition rules cannot. Because a relation path is just a sequence of the relation labels used as a feature, we can always use it, whether it satisfies temporal closure or not. In other words, temporal relation paths can provide partial clues for the cases that the strict transition rules do not work well.

Another reason stems from the characteristics of the TempEval relation labels. A GUI tool called TANGO[6] has functions dealing with inference rules. But there are only six temporal relations in TempEval: "BEFORE," "BEFORE-OR-OVERLAP," "OVERLAP," "OVERLAP-OR-AFTER," "AFTER," and "VAGUE." The six relations are coarser than Allen's 13 temporal relations. Strict inference rules such as "(a BEFORE b and b BEFORE c) $\Rightarrow$ (a BEFORE c)" are limited on a 6 x 6 relation matrix. Most inference rules such as "(a BEFORE-OR-OVERLAP b and b BEFORE-OR-OVERLAP c) $\Rightarrow$ (a {BEFORE, BEFORE-OR-OVERLAP, OVERLAP} c)" are not strict inference rules but narrowing rules. When using the result of the inferenced (narrowed) rules as features, we cannot include how the narrowing results are derived. Therefore, instead of making use of the results of the inference rules, we introduce the relation path features that are composed of the relations sequences of the inference steps. Actually in our experiments, this worked well on the TempEval data.

The other features we use for learning SVM classifiers are listed in the center column of Table 6.2. We divided them into two types: base features which are directly obtainable from the corpus (TempEval data); extended features which can be obtained

[6] `http://www.timeml.org/site/tango/tool.html`

Table 6.2: Used Features for Temporal Relation Identification

	SVM Pipeline			MLN Joint		
	Task A	Task B	Task C	Task A	Task B	Task C
EVENT-word	X		X	X		X
EVENT-POS	X		X	X		X
EVENT-stem	X		X	X		X
EVENT-aspect	X	X	X	X	X	X
EVENT-tense	X	X	X	X	X	X
EVENT-class	X	X	X	X	X	X
EVENT-polarity	X		X	X		X
TIMEX3-word	X			X		
TIMEX3-POS	X			X		
TIMEX3-value	X			X		
TIMEX3-type	X			X		
TIMEX3-TemporalFunction	X					
TIMEX3-FunctionInDocument	X					
TIMEX3-anchorTimeID	X					
TIMEX3-DCT order		X		X	X	
positional order	X			X		
in/outside	X			X		
unigram(word)	X		X	X		X
unigram(POS)	X		X	X		X
bigram(word)	X		X			
bigram(POS)	X		X	X		
trigram(word)	X		X			
trigram(POS)	X		X	X		X
Dependency-word	X	X	X	X	X	X
Dependency-POS	X	X	X	X	X	
Dependency-Path	X					
Temporal Relation Path	X		X			
Joint Formula				X	X	X

through pre-processing. Here we mention the features we used in detail.

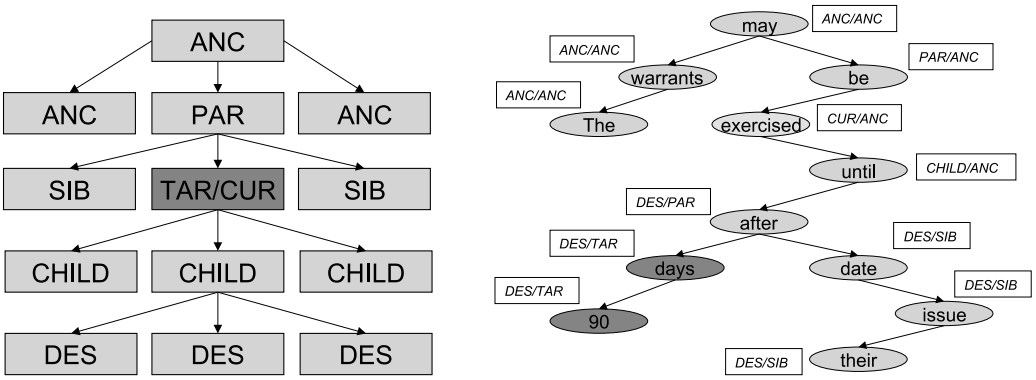
Base Features

Base features mainly cover the local information of time and event expressions.

EVENT expressions EVENT tags for event expressions have the following attributes: words (**EVENT-word**) (e.g., 'played', 'exercise'), aspect (**EVENT-aspect**) (e.g., 'PROGRESSIVE', etc.), polarity (**EVENT-polarity**) (e.g., 'POSITIVE' or 'NEGATIVE'), POS (**EVENT-POS**), stem (**EVENT-stem**), class (**EVENT-class**) (e.g., 'OCCURRENCE', 'REPORTING', etc.), and tense (**EVENT-tense**) (e.g., 'PAST', 'PRESENT', etc.) Tasks A and C use all of them. For Task B, we checked the effective features using the development set and only adopted class, tense and aspect.

TIME expressions TIMEX3 tags for time expressions have the following attributes: words (**TIMEX3-word**), value (**TIMEX3-value**), POS (**TIMEX3-POS**), type (**TIMEX3-type**) (e.g., 'DATE' 'TIME', etc.), TemporalFunction (**TIMEX3-TemporalFunction**) (e.g., 'true' or 'false', etc.), FunctionInDocument (**TIMEX3-FunctionInDocument**) (e.g., 'CREATION_TIME', 'EXPIRATION_TIME', etc.), and anchorTimeID (**TIMEX3-anchorTimeID**) (Time ID of the related time expression.) As in the case of EVENT, we used all of them for Task A. For Task B, the targets are only DCT and event expressions. So we focus on the time expressions that are in the target sentences and include the temporal order with DCT (**TIMEX3-DCT order**) (BEFORE, OVERLAP, or AFTER) as features. For Task C, we do not include any TIME informations.

Others For Task A, we used additional features because the task takes expressions only in the same sentences as the targets. We include the positional order of target expressions in the sentences (**positional order**) (not the temporal order but which expression comes first in the sentences). We also add word and POS unigrams based on the relative positions with the two expressions (**in/outside**): between them or outside of them. We used not only target time and event expressions but also time expressions in neighbor sentences. Unigrams, bigrams and trigrams of words and POS in the neighboring sentences are included for Task A and Task C (**unigram(word)**, **unigram(pos)**, **bigram(word)**, **bigram(pos)**, **trigram(word)**, and **trigram(pos)**). While explaining so many base features, we basically used as much information as possible.



(a) DepTree Position Labels (b) DepTree Position Labels (example)

Figure 6.8: Dependency Tree Position Labels

Extended Features

There are two main types of extended features: dependency relation labels with POS and joint relational features.

POS and dependency relation information The dependency information of the target sentences provides quite effective features, especially for Task A. We include the specific features concerning the positions where the time and event expressions appear in the dependency tree of the target sentence.

Figure 6.8 illustrates the dependency tree position labels. In this figure, we consider *exercised* as a current expression "CUR" and *90 days* as a target expression "TAR". In the dependency trees, for a word above the current (target) expression, we put an "ANC (ancestor)" label. For a word below the current (target), we label it with a "DES (descendant)." For a word that shares the same parent with the current (target), we label it with "SIB (sibling)" on the word. We also define "PAR (parent)" and "CHILD" but they are just specific cases of "ANC" and "DES."

The words in the same sentence with dependency tree positions and POS labels are used as features for Tasks A, B, and C (**Dependency-word**, **Dependency-POS**). For Task A, the relative position between target events and times on the dependency tree is also used as a feature (**Dependency-Path**).

Global Feature The last two columns in Table 6.2 represent global features. For SVM pipeline global features are above mentioned *temporal relation paths* (Figures

6.5, 6.6, and 6.7). For MLN joint they become *joint formulae* implementing global constraints of transition rules. The joint formulae will be presented in the next section.

6.3.2 MLN Joint Model

As stated before, our Markov Logic Model aims to jointly tackle Tasks A, B and C of the TempEval challenge. In this section we introduce the Markov Logic Network we designed for this goal.

Again, we have three *hidden* predicates, corresponding to Tasks A, B, and C: relE2T(e, t, r) represents the temporal relation of class r between an event e and a time expression t; relDCT(e, r) denotes the temporal relation r between an event e and DCT; relE2E($e1, e2, r$) represents the relation r between two events of the adjacent sentences, $e1$ and $e2$. Hidden predicates are summarized in Table 6.1.

Our observed predicates reflect information we were given (such as the words of a sentence), and additional information we extracted from the corpus (such as POS tags and parse trees). Note that the TempEval data also contained temporal relations that were not supposed to be predicted. These relations are represented using two observed predicates: relT2T($t1, t2, r$) for the relation r between two time expressions $t1$ and $t2$; dctOrder(t, r) for the relation r between a time expression t and a fixed DCT. An illustration of all *temporal* predicates, both hidden and observed, can be seen in Figure 6.3.

Local Formula

Our MLN is composed of several weighted formulae that we divide into two classes. The first class contains *local* formulae for the Tasks A, B and C. We say that a formula is local if it only considers the hidden temporal relation of a single event-event, event-time or event-DCT pair. The formulae in the second class are *global*: they involve two or more temporal relations at the same time, and consider Tasks A, B and C simultaneously.

The local formulae are based on features employed in previous work (Cheng et al., 2007; Bethard and Martin, 2007) and are listed in the right column of Table 6.2. What follows is a simple example in order to illustrate how we implement each feature as a formula (or set of formulae).

Table 6.3: Joint Formulae for Global Temporal Relation Identifier

Task	Formula
$A \rightarrow B$	$\text{dctOrder}(t, R1) \wedge \text{relE2T}(e, t, R2) \Rightarrow \text{relDCT}(e, R3)$
$B \rightarrow A$	$\text{dctOrder}(t, R1) \wedge \text{relDCT}(e, R2) \Rightarrow \text{relE2T}(e, t, R3)$
$B \rightarrow C$	$\text{relDCT}(e1, R1) \wedge \text{relDCT}(e2, R2) \Rightarrow \text{relE2E}(e1, e2, R3)$
$B, C \rightarrow B$	$\text{relDCT}(e1, R1) \wedge \text{relE2E}(e1, e2, R2) \Rightarrow \text{relDCT}(e2, R3)$
$A \rightarrow C$	$\text{relE2T}(e1, t1, R1) \wedge \text{relT2T}(t1, t2, R2) \wedge \text{relE2T}(e2, t2, R3) \Rightarrow \text{relE2E}(e1, e2, R4)$
$A, C \rightarrow A$	$\text{relE2T}(e2, t2, R1) \wedge \text{relT2T}(t1, t2, R2) \wedge \text{relE2E}(e1, e2, R3) \Rightarrow \text{relE2T}(e1, t1, R4)$

Consider the tense-feature for Task C. For this feature we first introduce a predicate $\text{tense}(e, t)$ that denotes the tense t for an event e. Then we add a set of formulae such as

$$\text{tense}(e_1, tns_1) \wedge \text{tense}(e_2, tns_2) \Rightarrow \text{relE2E}(e_1, e_2, r). \tag{6.2}$$

where e_1 and e_2 are variables for events, $tns1$ and $tns2$ are tense variables, and r is a temporal relation between e_1 and e_2.

For all possible combinations of tenses and temporal relations, Formula 6.2 is instantiated.[7] One of such instances is

$$\text{tense}(E32, \text{past}) \wedge \text{tense}(E33, \text{future}) \Rightarrow \text{relE2E}(E32, E23, \text{before}) \tag{6.3}$$

which represents a simple tense rule: an event with past tense occurs before the event with future tense. Actually this formula should acquired a high weight value.

Global Formula

Our global formulae are designed to enforce consistency between the three hidden predicates (and the two observed temporal predicates we mentioned earlier). They are based on the transition rules we briefly stated in Section 6.2.

Table 6.3 shows the set of formula templates we use to generate the global formulae. Here each template produces several instantiations. One example of a template instantiation is the following formula.

$$\text{dctOrder}(t1, \text{before}) \wedge \text{relDCT}(e1, \text{after}) \Rightarrow \text{relE2T}(e1, t1, \text{after}) \tag{6.4a}$$

[7] This type of "template-based" formulae generation can be performed automatically by the Markov Logic Engine.

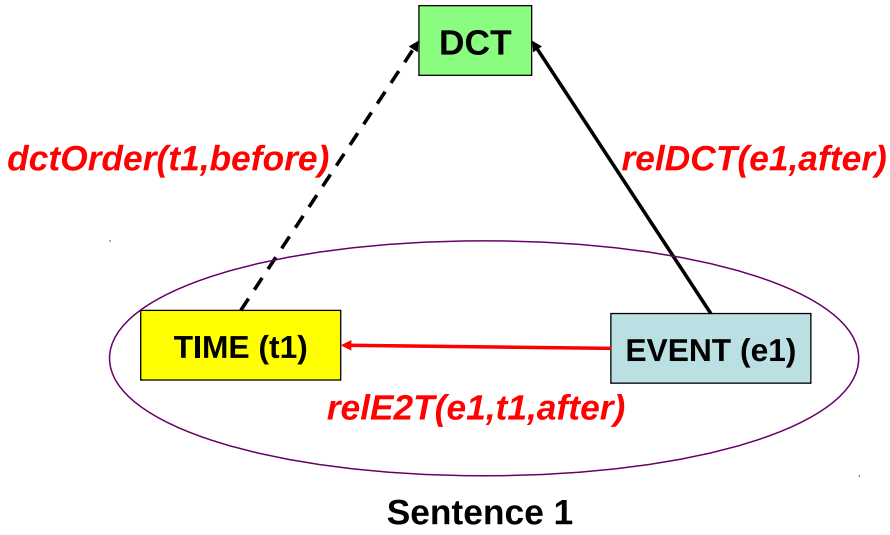


Figure 6.9: Temporal Transition Rule for Global Constraint 1

This formula is an expansion of the formula template in the second row of Table 6.3. We illustrate the three predicates of Formula 6.4a in Figure 6.9. Note that it utilizes the results of Task B to solve Task A.

Formula 6.4a should always hold[8], and hence we could easily implement it as a hard constraint in an ILP-based framework. However, some transition rules are less determinstic and should rather be taken as "rules of thumb". For example, formula 6.4b is a rule which we expect to hold often, but not always.

$$\text{dctOrder}(t1, \text{before}) \wedge \text{relDCT}(e1, \text{overlap}) \Rightarrow \text{relE2T}(e1, t1, \text{after}) \qquad (6.4b)$$

For the predicates of Formula 6.4b we also illustrate them in Figure 6.10. Fortunately, this type of soft rule poses no problem for Markov Logic: after training, Formula 6.4b will simply have a lower weight than Formula 6.4a. By contrast, in a "Local Classifier + ILP"-based approach as followed by Chambers and Jurafsky (2008b) it is less clear how to proceed in the case of soft rules. Surely it is possible to incorporate weighted constraints into ILPs, but how to learn the corresponding weights is not obvious.

[8]However, due to inconsistent annotations one will find violations of this rule in the TempEval data.

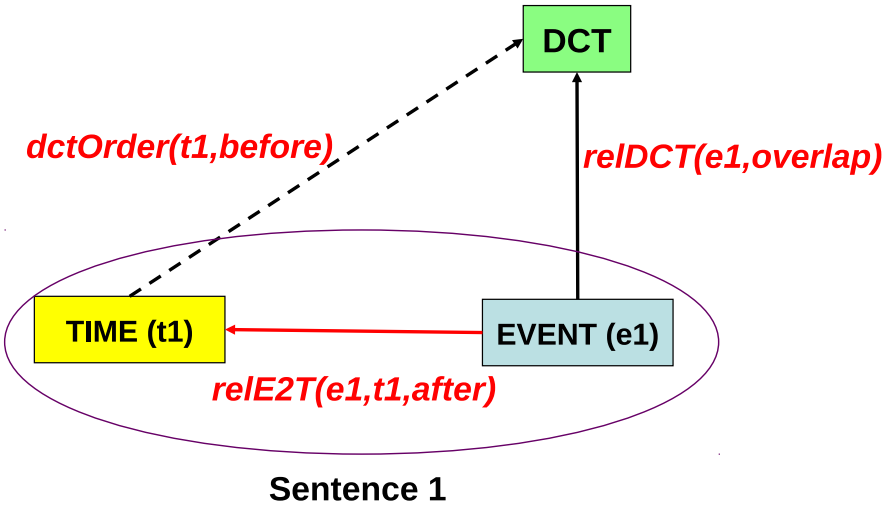


Figure 6.10: Temporal Transition Rule for Global Constraint 2

Table 6.4: Numbers of Labeled Relations for All Tasks in TempEval

	TRAIN	DEV	TEST	TOTAL
Task A	1359	131	169	1659
Task B	2330	227	331	2888
Task C	1597	147	258	2002

6.4 Experimental Setup

With our experiments we want to answer two questions: (1) does jointly tackling Tasks A, B, and C help to increase overall accuracy of temporal relation identification? (2) How does our approach compare to state-of-the-art results? In the following we will present the experimental set-up we chose to answer these questions.

In our experiments we use the test and training sets provided by the TempEval shared task. We further split the original training data into a training and a development set, used for optimizing parameters and formulae. For brevity we will refer to the training, development and test set as TRAIN, DEV and TEST, respectively. The numbers of temporal relations in TRAIN, DEV, and TEST are summarized in Table 6.4.

For feature generation we use the following tools.[9] POS tagging is performed with

[9]Since the TempEval trial has no restriction on pre-processing such as syntactic parsing, most participants used some sort of parsers.

TnT ver2.2;[10] for our dependency-based features we use MaltParser 1.0.0.[11] For inference in our models we use Cutting Plane Inference (Riedel, 2008) with ILP as a base solver. This type of inference is exact and often very fast because it avoids instantiation of the complete Markov Network. For learning we apply one-best MIRA (Crammer and Singer, 2003) with Cutting Plane Inference to find the current model guess. Both training and inference algorithms are provided by *Markov thebeast*,[12] a Markov Logic interpreter tailored for NLP applications.

Note that there are several ways to manually optimize the set of formulae to use. One way is to pick a task and then choose formulae that increase the accuracy for this task on DEV. However, our primary goal is to improve the performance of all the tasks together. Hence we choose formulae with respect to the total score over all three tasks. We will refer to this type of optimization as "averaged optimization". The total scores of the all three tasks are defined as follows:

$$\frac{C_a + C_b + C_c}{G_a + G_b + G_c}$$

where C_a, C_b, and C_c are the number of the correctly identified labels in each task, and G_a, G_b, and G_c are the numbers of gold labels of each task. Our system necessarily outputs one label to one relational link to identify. Therefore, for all our results, precision, recall, and F-measure are the exact same value.

For evaluation, TempEval proposed the two scoring schemes: "strict" and "relaxed". For strict scoring we give full credit if the relations match, and no credit if they do not match. On the other hand, relaxed scoring gives credit for a relation according to Table 6.5. For example, if a system picks the relation "AFTER" that should have been "BEFORE" according to the gold label, it gets neither "strict" nor "relaxed" credit. But if the system assigns "B-O (BEFORE-OR-OVERLAP)" to the relation, it gets a 0.5 "relaxed" score (and still no "strict" score).

6.5 Experimental Results

In the following we will first present our comparison of the local and global model. We will then go on to put our results into context and compare them to the state-of-the-art.

[10] http://www.coli.uni-saarland.de/~thorsten/tnt/

[11] http://w3.msi.vxu.se/~nivre/research/MaltParser.html

[12] http://code.google.com/p/thebeast/

Table 6.5: Evaluation Weights for Relaxed Scoring in TempEval

	B	O	A	B-O	O-A	V
B	1	0	0	0.5	0	0.33
O	0	1	0	0.5	0.5	0.33
A	0	0	1	0	0.5	0.33
B-O	0.5	0.5	0	1	0.5	0.67
O-A	0	0.5	0.5	0.5	1	0.67
V	0.33	0.33	0.33	0.67	0.67	1

B:	BEFORE	O:	OVERLAP
A:	AFTER	B-O:	BEFORE-OR-OVERLAP
O-A:	OVERLAP-OR-AFTER	V:	VAGUE

Table 6.6: Results on TEST Set in TempEval Task

model	task	Local		Global	
		strict	relaxed	strict	relaxed
SVM Pipeline	Task A	0.598	0.654	0.627	0.682
	Task B	0.758	0.775	0.758	0.775
	Task C	0.539	0.601	0.562	0.622
	All	0.648	0.689	0.662	0.702
MLN Joint	Task A	0.621	0.669	0.645	0.687
	Task B	0.737	0.753	0.758	0.777
	Task C	0.531	0.599	0.566	0.632
	All	0.641	0.682	0.668	0.708

6.5.1 Impact of Global Formulae

First, let us show the results on TEST in Table 6.6. You will find two columns, "Global" and "Local", showing scores achieved with and without global features or joint formulae, respectively. Clearly, the global models scores are higher than the local scores for all three tasks. This is also reflected by the last row of Table 6.6. Here we see that we have improved the averaged performance across the three tasks by approximately1.5% for SVM pipeline and 2.5% for MLN joint ($\rho < 0.01$, McNemar's test 2-tailed). Note that with 3.5% the improvements are particularly large for Task C of MLN joint. Comparing our two models with each other, MLN joint outperformed

Table 6.7: Runtime Comparison between SVM Pipeline vs MLN Joint in TempEval Task (sec.)

	SVM Task A	SVM Task B	SVM Task C	MLN
Train	163.3	80.0	107.1	43.3
Test	0.9	0.4	0.3	16.6

Table 6.8: Results with 10-fold Cross Validation for All in TempEval Task

	Local		Global	
model	strict	relaxed	strict	relaxed
SVM Pipeline	0.668	0.704	0.677	0.718
MLN Joint	0.667	0.707	0.689	0.727

SVM pipeline.

About the runtime comparison, we show the runtimes of SVM pipeline and MLN joint models in Table 6.7. The values in this table are calculated for "Global" models of SVM pipeline and MLN joint and they are averaged over three times running. SVM pipeline has a model for each task and we need to train it independently. In addition, fortunately, Tempeval data is smaller than the data for the previous two tasks. So, in this task, our MLN joint model quickly find the optimal solution and faster than SVM pipeline model at training time.

The TempEval test set is relatively small (see Table 6.4). Hence it is not clear how well our results would generalize in practice. To overcome this issue, we also evaluated the local and global models using 10-fold cross validation on the training data (TRAIN + DEV) and made sure the statistical significance. The corresponding results can be seen in Table 6.8. Note that the general picture remains: performances for the total scores of the three tasks are improved, and the the score is improved only slightly less than for the TEST results. However, the improvements for total scores are statistically significant ($\rho < 10^{-8}$, McNemar's test, 2-tailed).

To summarize, we have shown that by tightly connecting tasks A, B and C, we can improve temporal relation identification significantly. But are we just improving a weak baseline, or can joint modelling help to reach or improve the state-of-the-art results? We will try to answer this question in the next section.

Table 6.9: Comparison with Other Systems in TempEval

	Task A		Task B		Task C	
	strict	relaxed	strict	relaxed	strict	relaxed
TempEval Best	0.62	0.64	**0.80**	**0.81**	0.55	**0.64**
TempEval Average	0.56	0.59	0.74	0.75	0.51	0.58
CU-TMP	0.61	0.63	0.75	0.76	0.54	0.58
MLN Local Model	0.62	0.67	0.74	0.75	0.53	0.60
MLN Global Model	**0.65**	**0.69**	0.76	0.78	**0.57**	0.63
MLN Global Model (Task-Adjusted)	(0.66)	(0.70)	(0.76)	(0.79)	(0.58)	(0.64)

6.5.2 Comparison to the State-of-the-art

In this section we compare our results with the state-of-the-art results in TempEval. Since the performance of MLN joint is better than those of SVM pipeline, we choose the scores of MLN joint to represent our results.

In order to put our results into context, Table 6.9 shows them along those of other TempEval participants. In the first row, TempEval Best gives the best scores of TempEval for each task. Note that all but the strict scores of Task C are achieved by WVALI (Puscasu, 2007), a hybrid system that combines machine learning and hand-coded rules. In the second row we see the TempEval average scores of all six participants in TempEval. The third row shows the results of CU-TMP (Bethard and Martin, 2007), an SVM-based system that achieved the second highest scores in TempEval for all three tasks. CU-TMP is of interest because it is the best pure Machine-Learning-based approach so far.

The scores of our local and global model come in the fourth and fifth row, respectively. The last row in the table shows task-adjusted scores. Here we essentially designed and applied three global MLNs, each one tailored and optimized for a different task. Note that the task-adjusted scores are always about 1% higher than those of the single global model.

Let us discuss the results of Table 6.9 in detail. We see that for task A, our global model improves an already strong local model to reach the best results both for strict scores (with a 3% points margin) and relaxed scores (with a 5% points margin).

For Task C we see a similar picture: here adding global constraints helped to reach the best strict scores, again by a wide margin. We also achieve competitive relaxed

scores which are in close range to the TempEval best results.

Only for task B our results cannot reach the best TempEval scores. While we perform slightly better than the second-best system (CU-TMP), and hence report the best scores among all pure Machine-Learning based approaches, we cannot quite compete with WVALI.

6.6 Discussion

Let us discuss some further characteristics and advantages of our approach. First, notice that global formulae not only improve strict but also relaxed scores for all tasks. This suggests that we produce more ambiguous labels (such as BEFORE-OR-OVERLAP) in cases where the local model has been overconfident (and wrongly chose BEFORE or OVERLAP), and hence make less "fatal errors". Intuitively this makes sense: global consistency is easier to achieve if our labels remain ambiguous. For example, a solution that labels every relation as VAGUE is globally consistent (but not very informative).

Secondly, one could argue that our solution to joint temporal relation identification is too complicated. Instead of performing global inference, one could simply arrange local classifiers for the tasks into a pipeline. In fact, this has been done by Bethard and Martin (2007): they first solve task B and then use this information as features for Tasks A and C. While they do report improvements (0.7% on Task A, and about 0.5% on Task C), generally these improvements do not seem as significant as ours. What is more, by design their approach can not improve the first stage (Task B) of the pipeline.

On the same note, we also argue that our approach does not require more implementation efforts than a pipeline. Essentially we only have to provide features (in the form of formulae) to the Markov Logic Engine, just as we have to provide for a SVM or MaxEnt classifier.

Next, we show an example of our global model's improvement. Let us see a document with temporal annotations.

```
1 The official <EVENT eid="e290" mainevent="YES" class="
  REPORTING" aspect="NONE" tense="PRESENT" polarity="POS" stem=
  "add" pos="VERB">adds</EVENT>, though, that at the same time,
   we think he is someone who is capable of rational judgments
  when it comes to power.
2
3 And when he finds something unprofitable, then one can <EVENT
   eid="e300" mainevent="YES" class="PERCEPTION" aspect="NONE"
```

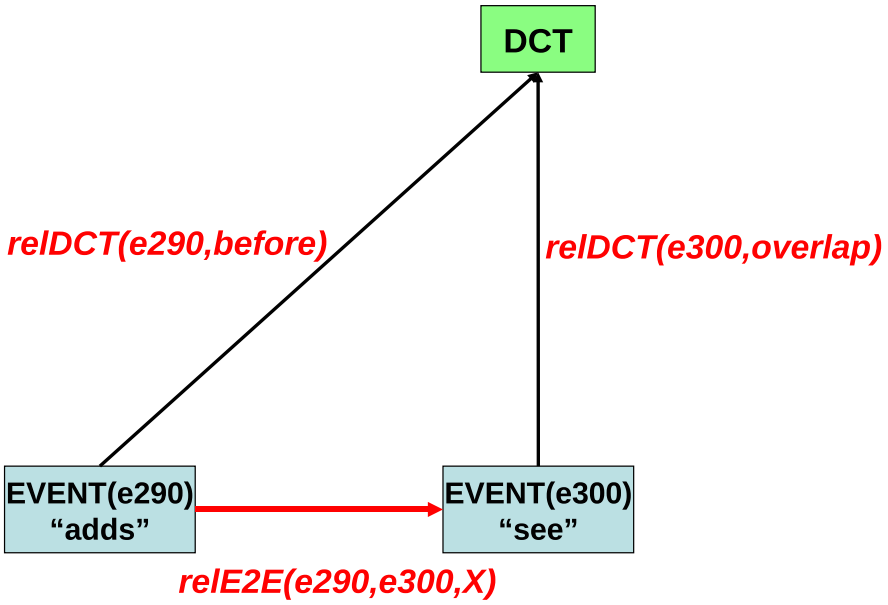


Figure 6.11: Success Example of Global Constraints in Temporal Relation Identification

```
tense="NONE" polarity="POS" stem="see" pos="VERB">see</EVENT>
 certain accommodations."

<TLINK relatedToTime="t397" eventID="e290" task="B" lid="153"
 relType="BEFORE"/>

<TLINK relatedToTime="t397" eventID="e300" task="B" lid="155"
 relType="OVERLAP"/>

<TLINK eventID="e290" task="C" lid="122" relType="OVERLAP"
relatedToEvent="e300"/>
```

We have three target relations between $e290$ and DCT; $e300$ and DCT; $e290$ and $e300$ as shown in Figure 6.11

Now let us focus on “$relE2E(e290, e300, X)$ (relation between $e290$ and $e300$)” which belongs to Task C. Our local model without global constraints identified it as “*after*” by mistake.

On the other hand, our global model yielded the transition rule,

$$relDCT(e290, before) \wedge relDCT(e300, overlap) \Rightarrow \neg relE2E(e290, e300, after) \quad (6.5)$$

which means “the relation between $e290$ and $e300$ is ambiguous, but at least “*after*”

is incorrect". This constraint is illustrated in Figure 6.11. Due to this constraint, our global model successfully identified the target "X" as the correct label "*overlap*".

Finally, it became more clear to us that there are problems inherent to this task and dataset that we cannot (or only partially) solve using global methods. First, there are inconsistencies in the training data that often mislead the learner as reflected by the low inter-annotator agreement – 72% for Tasks A and B, 68% for Task C. This problem applies to learning of local and global formulae/features alike.

Second, the training tagged data is relatively small (1659 for Task A, 2888 for B, 2002 for C). Obviously, this makes learning of reliable parameters more difficult, particularly when data is as noisy as in our case. Third, the temporal relations in the TempEval dataset only directly connect a small subset of events. This makes global formulae less effective.[13]

6.7 Summary

In this chapter we presented a novel approach to temporal relation identification. Instead of using local classifiers to predict temporal order in a pairwise fashion, our approach uses Markov Logic to incorporate both local features and global transition rules between temporal relations.

We have focused on transition rules between temporal relations of the three TempEval subtasks: temporal ordering of events, of events and time expressions, and of events and the document creation time. Our results have shown that global transition rules lead to significantly higher accuracy for all three tasks. Moreover, our global Markov Logic model achieves the highest scores reported so far for two of three tasks, and very competitive results for the remaining one.

While temporal transition rules can also be captured with an Integer Linear Programming approach (Chambers and Jurafsky, 2008b), Markov Logic has at least two advantages. First, handling of "rules of thumb" between less specific temporal relations (such as OVERLAP or VAGUE) is straightforward—we simply let the Markov Logic Engine learn weights for these rules. Second, there is less engineering overhead for us to perform, because we do not need to generate ILPs for each document.

However, potential for further improvements through global approaches seems to be limited by the sparseness and inconsistency of the data. To overcome this problem,

[13] See (Chambers and Jurafsky, 2008b) for a detailed discussion of this problem, and a possible solution for it.

it is effective to use external or untagged data along with methods for unsupervised learning in Markov Logic (Poon and Domingos, 2008).

Another direction of future work is multilingual temporal ordering such as TempEval-2[14] which has challenging temporal ordering tasks in five languages. Here we expect that while lexical and syntax-based features may be quite language dependent, global transition rules should hold across languages.

[14] http://www.timeml.org/tempeval2/

Chapter 7

Conclusion

7.1 Summary

This thesis has explored probabilistic logic approach to event structure analysis. Probabilistic logic approach allows us to combine humans' linguistic knowledges and statistical information acquired from corpora. We selected Markov Logic as one of probabilistic logic frameworks. Markov Logic is a framework which combines first order logic and Markov networks. With Markov Logic we can represent humans' knowledge by first-order logic and learning the validity of the formulae from corpora.

Events and their structures play important roles in natural language documents. Especially we focused our effort on event-argument structure analysis.

In Chapter 4, we targeted Japanese predicate-argument structure. Predicates are verbal and adverbial events and have arguments with semantic roles. Japanese predicates mainly have three types of roles: nominative (ga), accusative (wo), and dative (ni). We proposed Markov Logic model which jointly deals with the three types of roles. Our model could take all predicates in a same sentence into account simultaneously and find the most probable assignments of predicates and arguments. Our model achieved very competitive results without a large scale unlabeled data.

In Chapter 5, we tackled event structure analysis in biomedical domain which has distinctive characteristics. On biomedical corpora, we could also build a global joint model in a sentence. Furthermore, we incorporated coreference information to extract event-argument relations. This work suggested the effectiveness of exploiting coreference relations (argument-argument relations) for event extraction.

In Chapter 6, we focus on another aspect of event structures. We constructed analyzers which extract temporal relations for events. Since event is a change of state,

it generally has a temporal attribute. We analyzed temporal orders related to events. In order to prevent logical contradictions among several temporal relations, our model exploited global constraints based on temporal closures. As a result, we achieved state-of-the-art results on an English temporal annotated corpus.

Events and their structures we targeted in above chapters are essential elements to understand documents. By building sophisticated approaches to extract event structures, our work constructed a new foothold in document understanding. In the last section, we suggest some directions to explorer further progress.

7.2 Future Directions

The first strong direction is naturally expanding our approaches to semi-supervised ones. Incorporating unlabeled data is often effective for various tasks including event-argument relation extraction and temporal ordering.

However, a practical method of semi-supervised Markov Logic has not been proposed yet. Markov Logic approach is a framework which is computationally very hard even if we only utilize a small amount of labeled data. Thus, it is intractable to instantiate a large search space and find the most probable states with a large scale unlabeled data. Constructing a semi-supervised Markov Logic framework which can be practically used is the first priority and challenging task.

The second direction is jointly extracting event-argument relations and coreference relations. Though we have already exploited coreference relations to event-argument relation extractions in Chapter 5, event-argument relations also contributes to extract coreference relations. Since Japanese documents have many cross-sentential event-argument relations, event-argument relations help to extract coreference relations (Iida and Tokunaga, 2010).

One of th biggest issues in such a joint approach is time and space complexities. Usually, event-argument relation extraction are solved in a sentence-by-sentence manner. But to jointly extract event-argument and coreference relations, we have to solve problems in a document-by-document manner. Too large search space targeting a whole document often makes our problem infeasible. Therefore, in order to realize joint formulation, we must apply some algorithms of approximation and propose a model which can perform much more efficiently learning and inference.

Actually, the issues of complexity are very critical. If we overcome the barrier, we can find a path to the further goal. Let us see Figure 1.2 again. This figure denotes

not only event-argument relations but also anaphora and temporal relations. Jointly extracting event-argument and anaphora (coreference) relations is the problem to be solved at document-level. Accordingly, we can also expand the problem to joint extraction of event-argument, anaphora, and temporal relations. Jointly extracting these three types of relations corresponds to extract *event chains* and *anaphoric chains*. Chambers and Jurafsky proposed an approach to extracting such chains (Chambers and Jurafsky, 2008a; Chambers and Jurafsky, 2009). They tried to extract such chains from unlabeled data and the chains they obtained have general stories of the domain they targeted.

On the other hand, the chains we wanted to extract by our joint approach are not general but specific stories corresponding to each document. Of course, we assume to extract chains by supervised manner. Therefore, the chains we want to obtain have rich information and are useful to understand the corresponding document itself.

Though high level applications such as automatic document summarization and machine translation are dreams of Natural Language Processing, the automatic systems are still far from practical use. However, if event and anaphoric chains enable us to roughly understand the stories in a document, Automatic analyzers (summarizer or translator) will output results at practical level.

References

Takeshi Abekawa and Manabu Okumura. 2005. Corpus-based analysis of Japanese relative clause constructions. In *Proceedings of 2nd International Joint Conference on Natural Language Processing (IJCNLP)*, pages 46–57, Jeju Island, Korea.

James Allen. 1983. Maintaining knowledge about temporal intervals. In *Communications of the ACM*, pages 832–843.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*, ACL '98, pages 86–90, Stroudsburg, PA, USA. Association for Computational Linguistics.

Adam L. Berger, Stephen A. Della Pietra, and Vincent J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22:39–71.

Steven Bethard and James H. Martin. 2007. Cu-tmp: Temporal relation classification using syntactic and semantic features. In *Proceedings of the 4th International Workshop on SemEval-2007.*, pages 129–132.

Jari Björne, Juho Heimonen, Filip Ginter, Antti Airola, Tapio Pahikkala, and Tapio Salakoski. 2009. Extracting complex biological events with rich graph-based feature sets. In *BioNLP '09: Proceedings of the Workshop on BioNLP*, pages 10–18, Boulder, CO, USA. Association for Computational Linguistics.

Branimir Boguraev and Rie Kubota Ando. 2005. Timeml-compliant text analysis for temporal reasoning. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, pages 997–1003.

Ekaterina Buyko, Erik Faessler, Joachim Wermter, and Udo Hahn. 2009. Event extraction from trimmed dependency graphs. In *BioNLP '09: Proceedings of the Workshop on BioNLP*, pages 19–27, Boulder, CO, USA. Association for Computational Linguistics.

Nathanael Chambers and Dan Jurafsky. 2008a. Unsupervised learning of narrative event chains. In *Proceedings of ACL-08: HLT*, pages 789–797, Columbus, Ohio, June. Association for Computational Linguistics.

Nathanael Chambers and Daniel Jurafsky. 2008b. Jointly combining implicit constraints improves temporal ordering. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 698–706, Honolulu, Hawaii, October. Association for Computational Linguistics.

Nathanael Chambers and Dan Jurafsky. 2009. Unsupervised learning of narrative schemas and their participants. 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*, pages 602–610, Suntec, Singapore, August. Association for Computational Linguistics.

Yuchang Cheng, Masayuki Asahara, and Yuji Matsumoto. 2007. Naist.japan: Temporal relation identification using dependency parsed tree. In *Proceedings of the 4th International Workshop on SemEval-2007.*, pages 245–248.

James Clarke and Mirella Lapata. 2007. Modelling compression with discourse constraints. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1–11, Prague, Czech Republic, June. Association for Computational Linguistics.

Michael Collins. 2002. Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithms. In *EMNLP '02: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing*, pages 1–8, Morristown, NJ, USA. Association for Computational Linguistics.

Koby Crammer and Yoram Singer. 2003. Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3:951–991.

George Bernard Dantzig, D. R. Fulkerson, and Selmer Martin Johnson. 1954. *Solution of a Large-Scale Traveling-Salesman Problem*. Operations Research Society of America.

Hal Daumé III. 2004. Notes on cg and lm-bfgs optimization of logistic regression.

Pedro Domingos and Daniel Lowd. 2009. *Markov Logic: An Interface Layer for Artificial Intelligence*. Morgan and Claypool Publishers, 1st edition.

M. Ellsworth, K. Erk, P. Kingsbury, and S. Pado. 2004. PropBank, SALSA, and Framenet: How design determines product. In *Proceedings of LREC 2004, Workshop on Building Lexical Resources from Semantically Annotated Corpora*, Lisbon, Portugal.

Charles J. Fillmore, Charles Wooters, and Collin F. Baker. 2001. Building a large lexical databank which provides deep semantics. In *Proceedings of the Pacific Asian Conference on Language, Information and Computation(PACLIC)*, Hong Kong.

Michael R. Genesereth and Nils J. Nilsson. 1987. *Logical foundations of artificial intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Matthew Gerber and Joyce Y. Chai. 2010. Beyond nombank: a study of implicit arguments for nominal predicates. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1583–1592, Stroudsburg, PA, USA. Association for Computational Linguistics.

W.R. Gilks and DJ Spiegelhalter. 1996. *Markov chain Monte Carlo in practice*. Chapman & Hall/CRC.

Barbara J. Grosz, Scott Weinstein, and Aravind K. Joshi. 1995. Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, 21:203–225.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. 2009. The conll-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*, pages 1–18, Boulder, Colorado, June. Association for Computational Linguistics.

Trevor Hastie and Robert Tibshirani. 1998. Classification by pairwise coupling. In *The Annals of Statistics*, pages 507–513. MIT Press.

Magnus R. Hestenes and Eduard Stiefel. 1952. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, December.

Peter et al.(eds) Hühn. 2011. the living handbook of narratology.

Ryu Iida and Takenobu Tokunaga. 2010. Inter-sentential zero-anaphora resolution using shared arguments of predicate pairs. In *the 16th Annual Meeting of the Association for Natural Language Processing*, pages 804–807.

Ryu Iida, Mamoru Komachi, Kentaro Inui, and Yuji Matsumoto. 2007. Annotating a Japanese text corpus with predicate-argument and coreference relations. In *Proceedings of the Linguistic Annotation Workshop*, LAW '07, pages 132–139, Stroudsburg, PA, USA. Association for Computational Linguistics.

Satoru Ikehara, Masahiro Miyazaki, Satoshi Shirai, Akio Yokoo, Hiromi Nakaiwa, Kentaro Ogura, Yoshifumi Ooyama, and Yoshihiko Hayashi. 1997. *Nihongo Goi Taikei, A Japanese Lexicon*. Iwanami Shoten, Tokyo.

Kenji Imamura, Kuniko Saito, and Tomoko Izumi. 2009. Discriminative approach to predicate-argument structure analysis with zero-anaphora resolution. 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 (ACL-IJCNLP), Conference Short Papers*, pages 85–88, Suntec, Singapore, August. Association for Computational Linguistics.

Daisuke Kawahara and Sadao Kurohashi. 2004. Toward text understanding: Integrating relevance-tagged corpus and automatically constructed case frames. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC2004)*, pages 1833–1836.

Daisuke Kawahara and Sadao Kurohashi. 2006a. Case frame compilation from the web using high-performance computing. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC2006)*, pages 1344–1347.

Daisuke Kawahara and Sadao Kurohashi. 2006b. A fully-lexicalized probabilistic model for Japanese syntactic and case structure analysis. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 176–183, New York City, USA, June. Association for Computational Linguistics.

Daisuke Kawahara and Sadao Kurohashi. 2010. Acquiring reliable predicate-argument structures from raw corpora for case frame compilation. In Nicoletta

Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may. European Language Resources Association (ELRA).

Daisuke Kawahara, Sadao Kurohashi, and Koichi Hashida. 2002. Construction of Japanese relevance-tagged corpus (in Japanese). In *the 8th Annual Meeting of the Association for Natural Language Processing*, pages 495–498.

Jin-Dong Kim, Tomoko Ohta, and Jun'ichi Tsujii. 2008. Corpus annotation for mining biomedical events from literature. *BMC Bioinformatics*, 9(1):10+.

Jin-Dong Kim, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, and Jun'ichi Tsujii. 2009. Overview of bionlp'09 shared task on event extraction. In *BioNLP '09: Proceedings of the Workshop on BioNLP*, pages 1–9, Boulder, CO, USA. Association for Computational Linguistics.

Daphne Koller, 1999. *Probabilistic Relational Models*, pages 3–13. Springer, Berlin/Heidelberg, Germany.

Mamoru Komachi, Ryu Iida, Kentaro Inui, and Yuji Matsumoto. 2007. Learning based argument structure analysis of event-nouns in Japanese. In *Conference of the Pacific Association for Computational Linguistics (PACLING)*, pages 120–128, Melbourne, Australia, September.

Nada. Lavrac and Saso Dzeroski. 1994. *Inductive Logic Programming : Techniques and Applications*. Ellis Horwood, New York.

Dong C. Liu, Jorge Nocedal, Dong C. Liu, and Jorge Nocedal. 1989. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45:503–528.

J. W. Lloyd. 1987. *Foundations of Logic Programming; (2nd extended edition)*. Springer-Verlag New York, Inc., New York, NY, USA.

Inderjeet Mani, Marc Verhagen, Ben Wellner, Chong Min Lee, and James Pustejovsky. 2006. Machine learning of temporal relations. In *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting*

of the Association for Computational Linguistics, pages 753–760, Morristown, NJ, USA. Association for Computational Linguistics.

Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert Macintyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The penn treebank: Annotating predicate argument structure. In *ARPA Human Language Technology Workshop*, pages 114–119.

David McClosky and Eugene Charniak. 2008. Self-training for biomedical parsing. In *HLT '08: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies*, pages 101–104, Morristown, NJ, USA. Association for Computational Linguistics.

A. Meyers, R. Reeves, C. Macleod, R. Szekely, V. Zielinska, B. Young, and R. Grishman. 2004. The nombank project: An interim report. In A. Meyers, editor, *HLT-NAACL 2004 Workshop: Frontiers in Corpus Annotation*, pages 24–31, Boston, Massachusetts, USA, May 2 - May 7. Association for Computational Linguistics.

Ivan Meza-Ruiz and Sebastian Riedel. 2009a. Jointly identifying predicates, arguments and senses using Markov logic. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 155–163, Boulder, CO, USA, June. Association for Computational Linguistics.

Ivan Meza-Ruiz and Sebastian Riedel. 2009b. Multilingual semantic role labelling with Markov logic. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*, pages 85–90, Boulder, Colorado, June. Association for Computational Linguistics.

Raymond Ng and V. S. Subrahmanian. 1992. Probabilistic logic programming. *Inf. Comput.*, 101(2):150–201.

Martha Palmer, Paul Kingsbury, and Daniel Gildea. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31.

Hoifung Poon and Pedro Domingos. 2007. Joint inference in information extraction. In *Proceedings of the Twenty-Second National Conference on Artificial Intelligence*, pages 913–918, Vancouver, Canada. AAAI Press.

Hoifung Poon and Pedro Domingos. 2008. Joint unsupervised coreference resolution with Markov Logic. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 650–659, Honolulu, Hawaii, October. Association for Computational Linguistics.

Hoifung Poon and Lucy Vanderwende. 2010. Joint inference for knowledge extraction from biomedical literature. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 813–821, Los Angeles, California, June. Association for Computational Linguistics.

Georgiana Puscasu. 2007. Wvali: Temporal relation identification by syntactico-semantic analysis. In *Proceedings of the 4th International Workshop on SemEval-2007.*, pages 484–487.

James Pustejovsky, Jose Castano, Robert Ingria, Reser Sauri, Robert Gaizauskas, Andrea Setzer, and Graham Katz. 2003a. The timebank corpus. In *Proceedings of Corpus Linguistics 2003*, pages 647–656.

James Pustejovsky, Jose Castano, Robert Ingria, Reser Sauri, Robert Gaizauskas, Andrea Setzer, and Graham Katz. 2003b. Timeml: Robust specification of event and temporal expression in text. *IWCS-5, Fifth International Workshop on Computational Semantics.*

Adwait Ratnaparkhi. 1998. Maximum entropy models for natural language ambiguity resolution. Technical report, PhD thesis, University of Pennsylvania.

Matthew Richardson and Pedro Domingos. 2006. Markov logic networks. *Machine Learning*, 62(1-2):107–136.

Sebastian Riedel and James Clarke. 2006. Incremental integer linear programming for non-projective dependency parsing. In *EMNLP*, pages 129–137, Sydney, Australia, July. Association for Computational Linguistics.

Sebastian Riedel and Ivan Meza-Ruiz. 2008. Collective semantic role labelling with Markov logic. In *CoNLL 2008: Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 193–197, Manchester, England, August. Coling 2008 Organizing Committee.

Sebastian Riedel, Hong-Woo Chun, Toshihisa Takagi, and Jun'ichi Tsujii. 2009. A Markov logic approach to bio-molecular event extraction. In *BioNLP '09: Proceedings of the Workshop on BioNLP*, pages 41–49, Boulder, CO, USA. Association for Computational Linguistics.

Sebastian Riedel. 2008. Improving the accuracy and efficiency of map inference for Markov logic. In *Proceedings of UAI 2008*.

Dan. Roth and Wen tau. Yih. 2005. Integer linear programming inference for conditional random fields. In *Proc. of the International Conference on Machine Learning (ICML)*, pages 737–744.

Dan Roth. 1996. On the hardness of approximate reasoning. *Artif. Intell.*, 82:273–302, April.

Ryohei Sasano, Daisuke Kawahara, and Sadao Kurohashi. 2004. Automatic construction of nominal case frames and its application to indirect anaphora resolution. In *Proceedings of Coling 2004*, pages 1201–1207, Geneva, Switzerland, Aug 23–Aug 27. COLING.

Ryohei Sasano, Daisuke Kawahara, and Sadao Kurohashi. 2009. The effect of corpus size on case frame acquisition for discourse analysis. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 521–529, Boulder, Colorado, June. Association for Computational Linguistics.

Bart Selman, Henry Kautz, and Bram Cohen. 1993. Local search strategies for satisfiability testing. In *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 521–532.

Parag Singla and Pedro Domingos. 2005. Discriminative training of Markov logic networks. In *Proceedings of American Association for Artificial Intelligence*, pages 868–873.

Parag Singla and Pedro Domingos. 2006. Entity resolution with Markov logic. In *Proceedings of the Sixth IEEE International Conference on Data Mining*, pages 572–582, HongKong. IEEE Computer Society Press.

Noah A. Smith. 2004. Log-linear models.

Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544.

Jian Su, Xiaofeng Yang, Huaqing Hong, Yuka Tateisi, and Jun'ichi Tsujii. 2008. Coreference resolution in biomedical texts: a machine learning approach. In Michael Ashburner, Ulf Leser, and Dietrich Rebholz-Schuhmann, editors, *Ontologies and Text Mining for Life Sciences : Current Status and Future Perspectives*, number 08131 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany.

Hirotoshi Taira, Sanae Fujita, and Masaaki Nagata. 2008. A Japanese predicate argument structure analysis using decision lists. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 523–532, Honolulu, HI, USA. Association for Computational Linguistics.

Vladimir N. Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, USA.

Marc Verhagen, Robert Gaizaukas, Frank Schilder, Mark Hepple, Graham Katz, and James Pustejovsky. 2007. Semeval-2007 task 15: Tempeval temporal relation identification. In *Proceedings of the 4th International Workshop on SemEval-2007.*, pages 75–80.

J. Weston and C. Watkins. 1998. Multi-class support vector machines. Technical report, CSD-TR-98-04, Department of Computer Science, Royal Holloway, University of London, Egham, TW20 0EX, UK.

H. Paul. Williams. 1999. *Model Building in Mathematical Programming, 4th Edition*. Wiley, 4 edition, October.

Wayne L. Winston and Munirpallam Venkataramanan. 2003. *Introduction to Mathematical Programming: Applications and Algorithms*. Brooks/Cole; 4th Revised edition.

Fei Wu and Daniel S. Weld. 2008. Automatically refining the wikipedia infobox ontology. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 635–644, New York, NY, USA. ACM.

Xiaofeng Yang, Guodong Zhou, Jian Su, and Chew Lim Tan. 2004. Improving noun phrase coreference resolution by matching strings. In *Proceedings of 1st Internation Joint Conference of Natural Language Processing*, pages 326–333.

List of Publication

Journal Papers

1. Katsumasa Yoshikawa, Sebastian Riedel, Tsutomu Hirao, Masayuki Asahara, Yuji Matsumoto. Coreference Based Event-Argument Relation Extraction on Biomedical Text. *Journal of Biomedical Semantics*, Vol.2, Supplement 5, S6, October, 2011.

2. Katsumasa Yoshikawa, Sebastian Riedel, Tsutomu Hirao, Masayuki Asahara, Yuji Matsumoto. Coreference Based Event Extraction on Biomedical Text. *Transactions of the Japanese Society for Artificial Intelligence*, Vol.26, No.2, pp 318-323, January, 2011. (in Japanese)

3. Katsumasa Yoshikawa, Sebastian Riedel, Masayuki Asahara, Yuji Matsumoto. Joint Inference of Temporal Relation Identification with Markov Logic. *Transactions of the Japanese Society for Artificial Intelligence*, Vol.24, No.6, pp.521-530, September, 2009. (in Japanese)

International Conference

1. Katsumasa Yoshikawa, Masayuki Asahara, Yuji Matsumoto. Jointly Extracting Japanese Predicate-Argument Relation with Markov Logic. In *Proceedings of the 5th International Joint Conference on Natural Language Processing (IJCNLP)*, Chiang Mai, Thailand, November, 2011 (to appear).

2. Katsumasa Yoshikawa, Sebastian Riedel, Tsutomu Hirao, Masayuki Asahara, Yuji Matsumoto. Coreference Based Event-Argument Relation Extraction on Biomedical Text. In *Proceedings of the*

Fourth International Symposium on Semantic Mining in Biomedicine (SMBM), pp 93-101, UK, October, 2010.

3. Katsumasa Yoshikawa, Sebastian Riedel, Masayuki Asahara, Yuji Matsumoto. Jointly Identifying Temporal Relations with Markov Logic. In *Proceedings of Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and 25 the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP 2009)*, pp. 405-413, Singapore, August 2009.

Other Publications

1. Katsumasa Yoshikawa, Masayuki Asahara, Yuji Matsumoto. Jointly Extracting Japanese Predicate-Argument Relation with Markov Logic. In *Information Processing Society of Japan SIG Notes, NL-199-5*, pp.1-8, November, 2010. (in Japanese)

2. Katsumasa Yoshikawa, Sebastian Riedel, Tsutomu Hirao, Masayuki Asahara, Yuji Matsumoto. Coreference Based Event Extraction on Biomedical Text. In *Proceedings of the 24th Annual Conference of the Japanese Society for Artificial Intelligence (JSAI2010)*, 2C3-1, June, 2010. (in Japanese)

3. Katsumasa Yoshikawa, Sebastian Riedel, Masayuki Asahara, Yuji Matsumoto. Machine Learning on Temporal Relation Identification with Joint Inference. In *Japanese Society for Artificial Intelligence SIG Notes, FPAI-73*, pp. 61-67, March, 2009. (in Japanese)

4. Katsumasa Yoshikawa, Masayuki Asahara, Yuji Matsumoto. Jointly Extracting Japanese Predicate-Argument Relation with Markov Logic.

In *Information Processing Society of Japan SIG Notes, NL-187-5*, pp. 29-36, September, 2008. (in Japanese)

Awards

1. The Best Paper Award of the SMBM2010 (the Fourth International Symposium on Semantic Mining in Biomedicine), 2010. K. Yoshikawa, S. Riedel, T. Hirao, M. Asahara, Y. Matsumoto, Coreference based event-argument relation extraction on Biomedical Text.