NAIST-IS-DD1061203

**Doctoral Dissertation**

# Extracting Named Entity Relations from Large Text Corpora

Toru Hirano

September 18, 2012

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

A 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

Toru Hirano

Thesis Committee:

| | |
|---|---|
| Professor Yuji Matsumoto | (Supervisor) |
| Professor Satoshi Nakamura | (Co-supervisor) |
| Associate Professor Masashi Shimbo | (Co-supervisor) |
| Assistant Professor Mamoru Komachi | (Co-supervisor) |

# Extracting Named Entity Relations from Large Text Corpora*

Toru Hirano

**Abstract**

Much attention has recently been devoted to using the enormous amount of web text covering an exceedingly wide range of domains as a huge knowledge resource. To use web texts as knowledge resources, we need to extract information from texts, which are merely sequences of words, and convert them into a structured form. The aim of this thesis is to extract relation information between named entities because they provide key information about real-world entities and relations. This extracted information is critical for applications such as information retrieval and question answering.

In this thesis, we extract semantically-related named entity pairs, $X$ and $Y$, and their relations, $R$, from documents $D$ in structured form [$X$, $Y$, $R$, $D$]. For example, the relation information [Ichiro Yamada, Jiro Yamada, brother] should be extracted from the document, ID = $002$, "Ichiro Yamada, the Democratic Party, is Jiro Yamada's brother". In this example, the relation expression "brother" is explicitly appeared in the document. In contrast, there is relation information that no relation expressions are appeared in documents such as the relation information [Ichiro Yamada, the Democratic Party, member, $002$], extracted from the above document.

To extract both kinds of relations from documents, we decompose the relation extraction task into three tasks. The first is detecting

*Doctoral Dissertation, Department of Information Processing, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD1061203, September 18, 2012.

semantically-related pairs from named entity pairs that co-occur in a given document (**relation detection**). The second is recognizing a relation expression that demonstrates the explicit relation between the detected pair from the document (**relation expression recognition**), and the third one is estimating the relationship that exists between a detected pair that has an implicit relation (**relation estimation**).

The contributions of this thesis are follows. In relation detection task, the prior methods target only intra-sentential relation detection in which named entity pairs are located in the same sentence, in spite of the fact that many named entity pairs with semantic relations are inter-sentential. Our proposed method is a supervised learning method using contextual features for detecting a semantic relation between a given pair of named entities, which may be located in different sentences. In relation expression recognition task, to solve the problem that syntactic clues were rather infrequent in a number of samples, we propose a supervised learning method using two kinds of external information about candidates acquired from large text corpora automatically. One is lexical information with selected nouns that indicate relations. The other is relation predicting model which predicts present relations between named entities on the basis of past relations of the pair. We show that the proposed method outperformed the prior method through the experiments. In relation estimation task, using similarity measures of named entity pairs were rather infrequent in case of ambiguous named entities. To solve the problem, we propose the similarity measure combining similarity of named entity pairs and similarity of document in which the pair appeared.

**Keywords:**

relation extraction, named entity, relation detection, relation expression recognition, relation estimation

# 大規模テキストからの固有名詞間の関係抽出*

平野 徹

## 内容梗概

Web 上に存在する膨大なテキストは広い分野をカバーしており，巨大な知識源と考えることができる．テキストを知識源として活用するには，個々のテキストに含まれる情報を抽出し構造化された形式に変換する必要がある．本研究では，情報検索や質問応答などのアプリケーションにおいて重要な知識源となる実世界の実体を指し示す固有名詞間の関係情報を抽出することを目的とする．

本研究で抽出する関係情報は，個々の文書 ($D$) で言及されている意味的な関係のある固有名詞の組 ($X$,$Y$) とその間の関係 ($R$) を［$X$, $Y$, $R$, $D$］の構造化された形で表現した情報である．

例えば,「民主党の山田一郎は山田次郎の兄です」の文書 ($ID = 002$) から抽出される関係情報は［山田一郎，山田次郎，兄，002］がある．この例では，固有名詞間の関係を表す表現「兄」が文書中に明記されているが，明記されていない関係情報も本研究では抽出対象とする．例えば，上の文書 ($ID = 002$) からは，関係を示す表現が文書中に明記されていないが「民主党」と「山田一郎」の間に「党員」の関係があると読み取れるため，関係情報［山田一郎，民主党，党員，002］も抽出できる．

上記の 2 種類の関係情報を抽出するために，本研究では，(1) 入力文書内で共起する固有名詞の組から何らかの関係を有する組を選択 (**関係性判定**) し，(2) 選択された組がどういう関係にあるのかを示す表現を入力文書から抽出 (**関係表現同定**) し，(3) 関係を示す表現が文書中に存在しない組の関係を推定 (**関係推定**) する，3 つのタスクに分ける．本研究の貢献は以下のとおりである．(1) 関係性判定タスクは，従来，同一文内で共起する組に対して関係の有無を判定することはできたが，日本語において頻出する文をまたいで共起する組に対して判定することはできなかった．そこで，照応解析で用いられている文脈的情報を関係性判定タ

*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-DD1061203, 2012 年 9 月 18 日.

スクに適用し文をまたいで共起する組に対しても判定可能な手法を提案した．(2) 関係表現同定タスクでは，文の構造情報だけでは関係表現か判断できない事例に対して，大規模テキストから自動獲得した 2 種類の外部知識を利用する手法を提案した．1 つは関係を示す名詞を推定した語彙的知識で，もう 1 つは対象組の過去の関係から現在の関係を予測する関係予測モデルである．評価実験では提案した 2 種類の外部知識を用いることの有効性を確認した．(3) 関係推定タスクにおいては，関係情報の類似性に基づく推定手法の根幹を担う類似度尺度について，上記 (1)(2) のタスクで自動獲得された関係情報に基づく固有表現組 ($X$,$Y$) の類似度と固有表現組の出力する文書 ($D$) の類似度を組み合わせた尺度を提案した．評価実験では 2 種類の類似度の混合割合を変えた実験を実施し，提案手法の有効性を確認した．

**キーワード**

関係抽出, 固有名詞, 関係性判定, 関係表現同定, 関係推定

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

CHAPTER

# 1

# INTRODUCTION

Much attention has recently been devoted to using the enormous amount of web text covering an exceedingly wide range of domains as a huge knowledge resource. To use web texts as knowledge resources, we need to extract information from texts, which are merely sequences of words, and convert them into a structured form. Although extracting information from texts in a structured form is difficult, relation extraction is one approach that makes it possible to use web texts as knowledge resources.

The aim of this thesis is to extract relation information between named entities because they provide key information about real-world entities and relations. This extracted information is critical for applications such as information retrieval, question answering, and the construction of an ontology (Zhu et al., 2009; Wong et al., 2010).

In this thesis, we extract semantically-related named entity pairs, $X$ and $Y$, and their relations, $R$, from Japanese documents $D$ in struc-

tured form [$X$, $Y$, $R$, $D$]. For example, the relation information [Ichiro Yamada$_1$, Japan$_2$, prime minister$_3$, 001] should be extracted from the following document, ID = 001. In the example, the numbers show correspondences of words between Japanese and English.

**document id = 001**

Yamada Ichiro$_1$-wa Nihon$_2$-no shusho$_3$-desu.
( Ichiro Yamada$_2$ is the prime minister$_3$ of Japan$_2$.)

It is possible to say that all named entity pairs that co-occur within a document are semantically related in some way. Conforming to the guidelines of relation extraction in English, Relation Detection and Characterization, used in the Automatic Content Extraction program[1], we state that two named entities that co-occur within a document are semantically related if the document can be read as demonstrating a relation that satisfies either of the following rules between the pair:

- One entity is an attribute value of the other
- Both entities are arguments of the same predicate

We divide the relation information extracted by following the above definition into two types, explicit and implicit relations, on the basis of relation $R$. An explicit relation means that there is an expression that indicates the relation between the named entity pair in the given document, while an implicit relation means that there is no such expression. For example, the relation information [Ichiro Yamada$_2$, Jiro Yamada$_3$, brother$_4$, 002], extracted from the document, ID = 002, demonstrates an explicit relation. In contrast, the relation information [Ichiro Yamada$_2$, the Democratic Party$_1$, member, 002], extracted from the same document, exhibits an implicit relation because no expression directly indicates the relation (e.g. member) between "Ichiro Yamada" and "the Democratic Party" in the document.

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

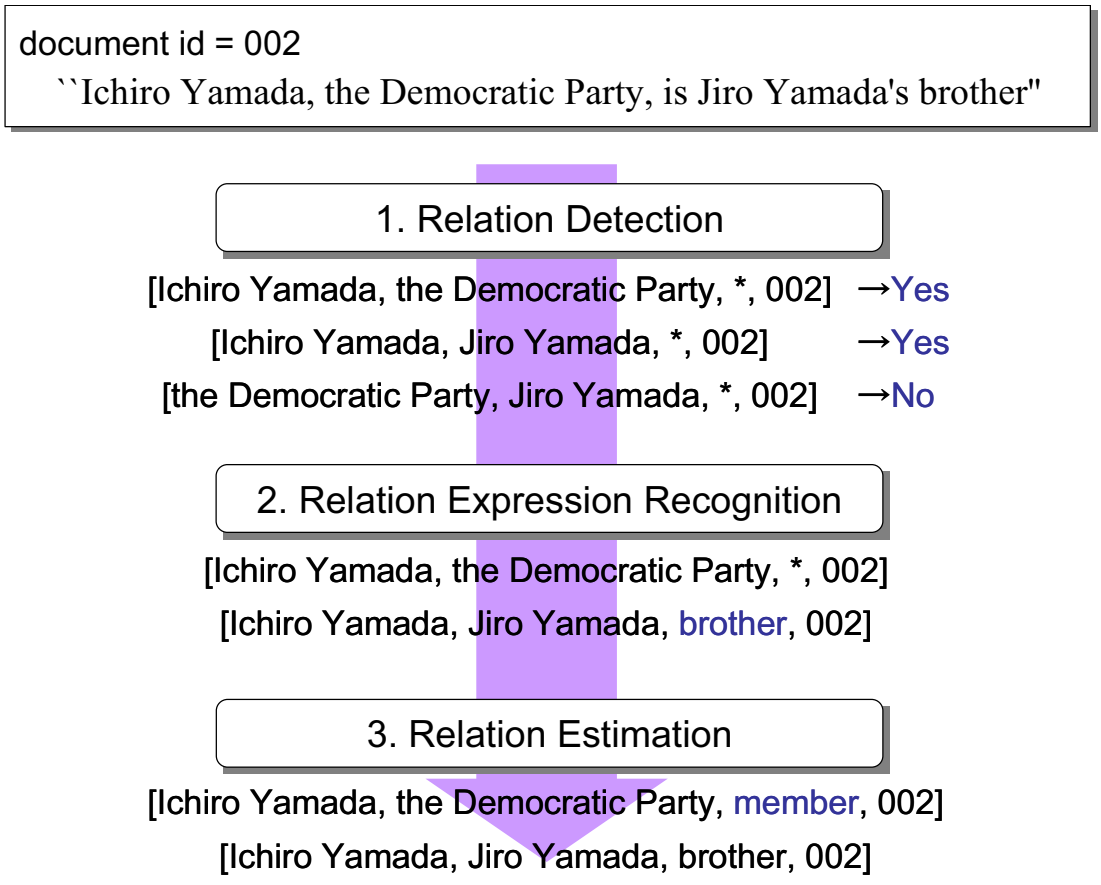


Figure 1.1. Overall view of the three tasks for relation extraction

**document id = 002**

Minshuto$_1$-no Yamada Ichiro$_2$-wa Yamada Jiro$_3$-no ani$_4$-desu.

( Ichiro Yamada$_2$, the Democratic Party$_1$, is Jiro Yamada$_3$'s brother$_4$. )

To extract both explicit and implicit relations from documents, we decompose the relation extraction task into three tasks. The first is detecting semantically-related pairs from named entity pairs that co-occur in a given document (**relation detection**). The second is recognizing a relation expression that demonstrates the explicit relation between the detected pair from the document (**relation expression recognition**), and the third one is estimating the relationship that

exists between a detected pair that has an implicit relation (**relation estimation**). Figure 1.1 shows that overall view of the three tasks with the example of the document, ID = 002. In relation detection task, the pairs [Ichiro Yamada, Jiro Yamada] and [Ichiro Yamada, the Democratic Party] should be detected as semantically-related ones and [the Democratic Party, Jiro Yamada] should not. Then, in relation expression recognition task, "brother" should be recognized as relation expression for [Ichiro Yamada, Jiro Yamada] and no expression should be recognized for [Ichiro Yamada, the Democratic Party]. At last, in relation estimation task, the relationship "member" should be estimated for [Ichiro Yamada, the Democratic Party]. In the tasks of relation expression recognition and relation estimation, we have to decide not only the relation between the pair but also which named entity in the pair is subject, $X$, of the relation.

The contributions of this thesis are follows. In relation detection task, various supervised learning approaches have been explored (Zelenko et al., 2003; Kambhatla, 2004; Culotta and Sorensen, 2004). They use two kinds of features: syntactic ones and word-based ones, for example, the path of the given pair in the parse tree and the word n-gram between named entities (Kambhatla, 2004). They target only intra-sentential relation detection in which named entity pairs are located in the same sentence, in spite of the fact that about 43.6% of named entity pairs with semantic relations are inter-sentential in Japanese documents. Our proposed method is a supervised learning method using contextual features for detecting a semantic relation between a given pair of named entities, which may be located in different sentences.

In relation expression recognition task, to recognize a relation expression from words located between a given pair in English, the prior work proposed methods using conditional random fields or Markov logic networks using only word-based features (Banko and Etzioni, 2008; Zhu et al., 2009). In a preliminary observation, we found that word-based features, even syntactic ones, were rather infrequent in a

number of samples. To solve the above problem, we propose a supervised learning method using two kinds of external information about candidates acquired from large text corpora automatically. One is lexical information with selected nouns that indicate relations. The other is relation predicting model which predicts present relations between named entities on the basis of past relations of the pair. We show that the proposed method outperformed the prior method through the experiments in Japanese corpus.

In relation estimation task, previous work used the idea that similar noun pairs must have the same relations to estimate implicit relation between not only named entities but also general nouns (Shimazu et al., 1986; Kurohashi and Sakai, 1999; Srikumar et al., 2008). To calculate similarity of named entities, several similarity measures of named entities on the basis of extracted huge relational information have been proposed for the purpose of paraphrasing or selective preference (Lin and Pantel, 2001; Hasegawa et al., 2004; Bollegala et al., 2010; Ritter et al., 2010). However, to estimate implicit relations, these similarity measures were rather infrequent in case of ambiguous named entities because they actually use only a named entity in the pair to disambiguate the other named entity. To solve the problem, we propose the similarity measure combining similarity of named entity pairs and similarity of document in which the pair appeared.

The rest of this thesis is organized as follows. We describe related work of relation extraction in Chapter 2. In Chapter 3, we propose a supervised learning method using contextual features for detecting a semantic relation between a given pair of named entities, which may be located in different sentences. Chapter 4 presents our proposed supervised learning method using two kinds of external information about candidates, lexical information and relation predicting model, acquired from large text corpora automatically to recognize relation expression of a given pair. In Chapter 5, we propose the similarity measure combining similarity of named entity pairs and similarity of document in which the pair appeared to estimate implicit relations

of named entity pairs. Finally, Chapter 6 concludes our work and presents the future directions.

CHAPTER

# 2

# RELATED WORK

## 2.1. Information extraction

Relation extraction task is a kind of information extraction task. Here we present related work on information extraction.

Information Extraction refers to the automatic extraction of structured information such as entities, relationships between entities, and attributes describing entities from unstructured sources. Early extraction tasks were concentrated around the identification of named entities, like people and company names and relationship among them from natural language text. The scope of this research was strongly influenced by two competitions, the Message Understanding Conference (MUC) and Automatic Content Extraction (ACE) program.

The advent of the Internet considerably increased the extent and diversity of applications depending on various forms of information extraction. Applications such as comparison shopping, and other au-

tomatic portal creation applications, lead to a frenzy of research and commercial activity on the topic. As society became more data oriented with easy online access to both structured and unstructured data, new applications of structure extraction came around.

Structure extraction is useful in a diverse set of applications. We list a representative subset of these.

## News Tracking

A classical application of information extraction, which has spurred a lot of the early research, is automatically tracking specific event types from news sources. The popular MUC and ACE competitions are based on the extraction of structured entities like people and company names, and relations such as " is-CEO-of " between them. Other popular tasks are: tracking disease outbreaks (Grishman et al., 2002), and terrorist events from news sources. Consequently there are several research publications (Grishman, 1997; Turmo et al., 2006) and many research prototypes (Hobbs et al., 1993; Riloff, 1993; Cunningham et al., 2002; Grishman et al., 2002) that target extraction of named entities and their relationship from news articles. Two recent applications of information extraction on news articles are: the automatic creation of multimedia news by integrating video and pictures of entities and events annotated in the news articles, and hyperlinking news articles to background information on people, locations, and companies.

## Customer Care

Any customer-oriented enterprise collects many forms of unstructured data from customer interaction; for effective management these have to be closely integrated with the enterprise 's own structured databases and business ontologies. This has given rise to many interesting extraction problems such as the identification of product names and product attributes from customer emails, linking of customer emails to a specific transaction in a sales database (Chakaravarthy et al.,

2006; Bhide et al., 2007), the extraction of merchant name and addresses from sales invoices (Zhu et al., 2007), the extraction of repair records from insurance claim forms (Popowich, 2005), the extraction of customer moods from phone conversation transcripts (Jansche and Abney, 2002), and the extraction of product attribute value pairs from textual product descriptions (Ghani et al., 2006).

## Personal information management

Personal information management (PIM) systems seek to organize personal data like documents, emails, projects and people in a structured inter-linked format (Cai et al., 2005; Chakrabarti et al., 2005; Cutrell and Dumais, 2006). The success of such systems will depend on being able to automatically extract structure from existing predominantly file-based unstructured sources. Thus, for example we should be able to automatically extract from a PowerPoint file, the author of a talk and link the person to the presenter of a talk announced in an email. Emails, in particular, have served as testbeds for many extraction tasks such as locating mentions of people names and phone numbers (Minkov et al., 2005), and inferring request types in service centers (Cohen et al., 2005).

## Scientific Applications

The recent rise of the field of bio-informatics has broadened the scope of earlier extractions from named entities, to biological objects such as proteins and genes. A central problem is extracting from paper repositories such as Pubmed, protein names, and their interaction (Bunescu et al., 2005). Since the form of entities like Gene and Protein names is very different from classical named entities like people and companies, this task has helped to broaden the techniques used for extraction.

### Opinion Databases

There are innumerable web sites storing unmoderated opinions about a range of topics, including products, books, movies, people, and music. Many of the opinions are in free text form hidden behind Blogs, newsgroup posts, review sites, and so on. The value of these reviews can be greatly enhanced if organized along structured fields. For example, for products it might be useful to find out for each feature of the product, the prevalent polarity of opinion (Liu et al., 2005; Popescu and Etzioni, 2005).

### Comparison Shopping

There is much interest in creating comparison shopping web sites that automatically crawl merchant web sites to find products and their prices which can then be used for comparison shopping (Doorenbos et al., 1997). As web technologies evolved, most large merchant web sites started getting hidden behind forms and scripting languages. Consequently, the focus has shifted to crawling and extracting information from form-based web sites (He et al., 2007).

## 2.2. Relation Extraction

### 2.2.1 Both Explicit and Implicit Relations

The "Message Understanding Conference" and "Automatic Content Extraction" programs have tackled relation extraction. The goal was to extract predefined semantic relations, $R$, of named entity pairs, $X$ and $Y$, from a document, $D$ in structured form [$X$, $Y$, $R$, $D$]. Various supervised learning approaches have been explored to date (Zelenko et al., 2003; Kambhatla, 2004; Culotta and Sorensen, 2004). They use two kinds of features: syntactic ones and word-based ones, for example, the path of the given pair of named entities in the parse tree and the

Table 2.1. Relation Types on Automatic Content Extraction Programs

| Type | Subtype | Argument $X$ | Argument $Y$ |
|---|---|---|---|
| Physical | Located | PER | FAC, LOC, GPE |
| | Near | PER, FAC, GPE, LOC | FAC, GPE, LOC |
| Part-whole | Geographical | FAC, LOC, GPE | FAC, LOC, GPE |
| | Subsidiary | ORG | ORG, GPE |
| Personal-Social | Business | PER | PER |
| | Family | PER | PER |
| | Lasting-Personal | PER | PER |
| ORG-Affiliation | Employment | PER | ORG, GPE |
| | Ownership | PER | ORG |
| | Founder | PER, ORG | ORG, GPE |
| | Student-Alum | PER | ORG |
| | Sports-Affiliation | PER | ORG |
| | Investor-Shareholder | PER, ORG, GPE | ORG, GPE |
| | Membership | PER, ORG, GPE | ORG |
| Agent-Artifact | UOIM | PER, ORG, GPE | FAC |
| Gen-Affiliation | CRRE | PER | PER, LOC, GPE,ORG |
| | Org-Location-Origin | ORG | LOC, GPE |

word n-gram between named entities with Maximum Entoropy Model (Kambhatla, 2004).

According to the latest guidelines of "Automatic Content Extraction" program, there are 6 types and 17 subtypes of relations are predefined. Table 2.1 shows the detail of the predefined relations. They also defined permitted relation arguments. For example, relation "Physical.Located" is permitted Person in argument $X$, and Facility, Location, and Geo-political Entity in argument $Y$.

With the guidelines and the annotated corpus, the task of relation extraction is designed to classify named entity pairs co-occurred in a given document into the 17 + 1 (no relation between the pair) relation subtypes. In this way, it can extract both explicit and implicit relation information from documents. However, it can not extract more precise relation information even relation expressions explicitly appeared

in the document. For example, even there is a relation expression "brother" of the pair in the document, extracted relation information is "Personal-Social.Family".

There is another line of related work (Brin, 1998; Agichtein and Gravano, 2000; Pantel and Pennacchiotti, 2006), bootstrapping methods, to extract given semantic relations, $R$, of named entity pairs, $X$ and $Y$, from a document, $D$, in structured form [$X$, $Y$, $R$, $D$]. Bootstrapping is a general framework for reducing the requirement of manual annotation. Therefore, bootstrapping methods solve the disadvantage of supervised methods which we described above.

Bootstrapping methods alternate two phases, pattern extraction and instance extraction. For example, suppose we would like to extract "brother" relations between named entities. In pattern extraction phase, patterns like "$X$ is brother of $Y$" which co-occur frequently with a seed such as [Ichiro Yamada, Jiro Yamada] will be selected from a corpus. Confidence score is ssigned to each pttern depending on co-ocuurrence strangth to seed instances. Only top highest $k$ patterns are selected. It is necessary to assign low scores to generic patterns and high scores to patterns with high relatedness to the seed instances.

In instance extraction phase, on the other hand, new instances like [Taro Suzuki, Jiro Suzuki] which co-occur with the patterns will be acquired and used for the next iteration. Compute confidence scores of enumeated instances and select high-confidence instances to add to the seed instance set. It is desirable to keep only high-confidence instances at this phase, as they are used as seed instances for the next iteration.

Bootstrapping itrates the above two phases several times until stopping criiteria are met. Acquired instances tend to become noisy as the iteration proceeds, so it is important to terminate before semantic drift ocuurs.

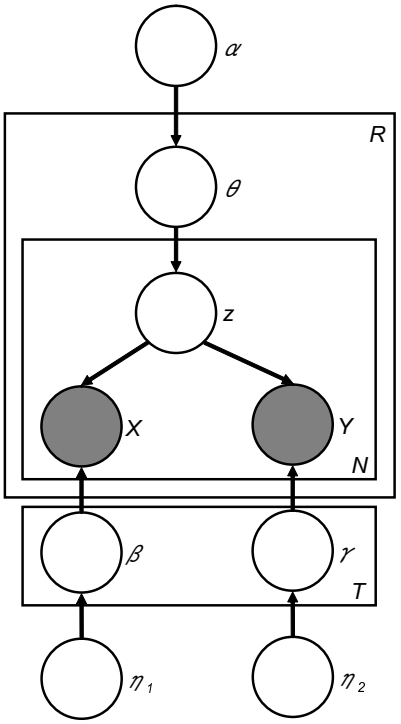


Figure 2.1. JointLDA

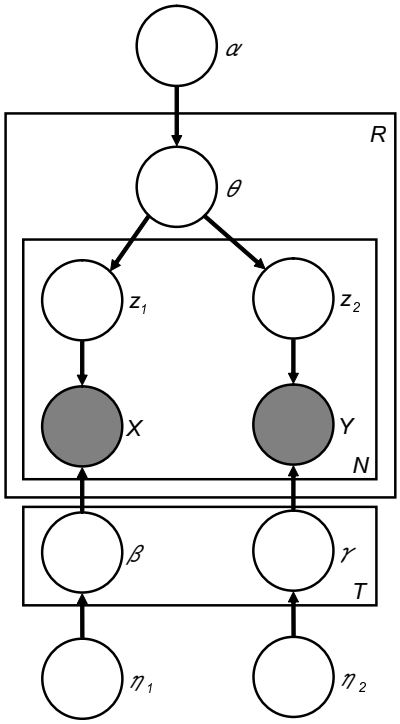


Figure 2.2. LinkLDA

### 2.2.2 Only Implicit Relations

Several supervised approaches using similarity of noun phrases have been proposed to estimate implicit relations between noun phrases to date (Shimazu et al., 1986; Kurohashi and Sakai, 1999; Srikumar et al., 2008). As implicit relation types, Shimazu et al. (1986) defined about 80 relation types, such as possession, whole-part, purpose or instruments. While, Kurohashi and Sakai (1999) defined only five relation types, such as obligate cases or possession. Even though definitions of implicit relation types vary, the common idea of these methods is that similar noun phrase pairs have the same relations. For instance, Kurohashi and Sakai (1999) first identified the class of noun phrases using a thesaurus and then estimated implicit relation types between pairs of identified classes.

To calculate similarity of named entities, several similarity mea-

sures of named entities have been proposed (Lin and Pantel, 2001; Hasegawa et al., 2004; Bollegala et al., 2010; Ritter et al., 2010), such as DIRT and LinkLDA, on the basis of extracted explicit relations for the purpose of paraphrasing or selectional preference. As the similarity measure of named entities, the state-of-the-art similarity measure was proposed by Ritter et al. (2010) with LinkLDA framework.

Ritter et al. (2010) presents a series of topic models for the task of computing selectional preferences. These models vary in the amount of independence they assume between $X$ and $Y$. At one extreme is IndependentLDA, a model which assumes that both $X$ and $Y$ are generated completely independently. On the other hand, JointLDA, the model at the other extreme assumes both arguments of a specific extraction are generated based on a single hidden variable $Z$. LinkLDA lies between these two extremes and LinkLDA was reported as the best model for extracted relation information. Comparing JointLDA and LinkLDA, instead of imposing a hard constraint that $z_1 = z_2$ in JointLDA, LinkLDA simply assigns a higher probability to states in which $z_1 = z_2$, because both hidden variables are drawn from the same (sparse) distribution $\theta_r$.

### 2.2.3 Only Explicit Relations

Recently, open information extraction, a novel domain-independent extraction paradigm, has been suggested (Shinyama and Sekine, 2006; Banko and Etzioni, 2008).

Several clustering-based methods have been proposed to extract semantically related named entity pairs, $X$ and $Y$, and their relation expressions, $R$, from a large corpus in structured form [$X$, $Y$, $R$] without predefined relations (Hasegawa et al., 2004; Shinyama and Sekine, 2006; Bollegala et al., 2010). These methods extract multiple words around named entities in a document as features and cluster named entity pairs based on distributional similarity of the features. The methods output named entity pairs belonging to the same clusters that have the same relations. They also output key features of the cluster as

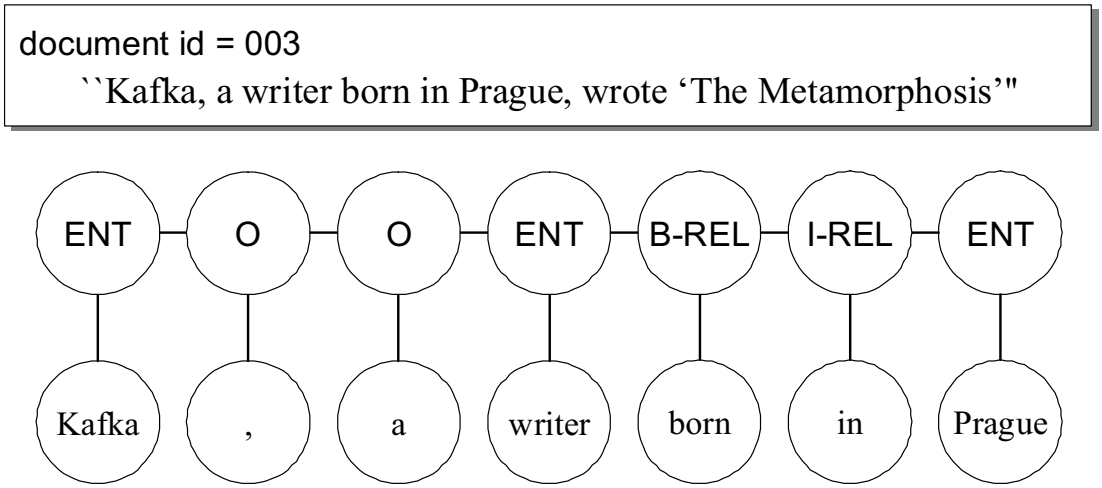


Figure 2.3. Relation Extraction as Sequence Labeling: A CRF is used to identify the relationship, "born in", between "Kafka" and "Prague"

relation expressions. These clustering-based methods extract relations between named entities from a corpus rather than a document.

There is another line of related work (Banko and Etzioni, 2008; Zhu et al., 2009), self-supervised or bootstrapping methods, to extract relation expressions, $R$, between a named entity pair, $X$ and $Y$, from a given document, $D$, in structured form [$X$, $Y$, $R$, $D$] without predefined relations. The idea of these methods is to discover domain independent extraction patterns.

Banko proposed a self-supervised learning method using conditional random fields to extract a relation expression from words located between a given pair (Banko and Etzioni, 2008). Each pair of named entities appearing no more than a maximum number of words apart and their surrounding context are considered as possible evidence for relation extraction. The named entity pair serves to anchor each end of a linear-chain CRF, and both named entities in the pair are assigned a fixed label of ENT. Tokens in the surrounding context are treated as possible textual cues that indicate a relation, and can be assigned one of the following labels: B-REL, indicating the start of a relation, I-REL,

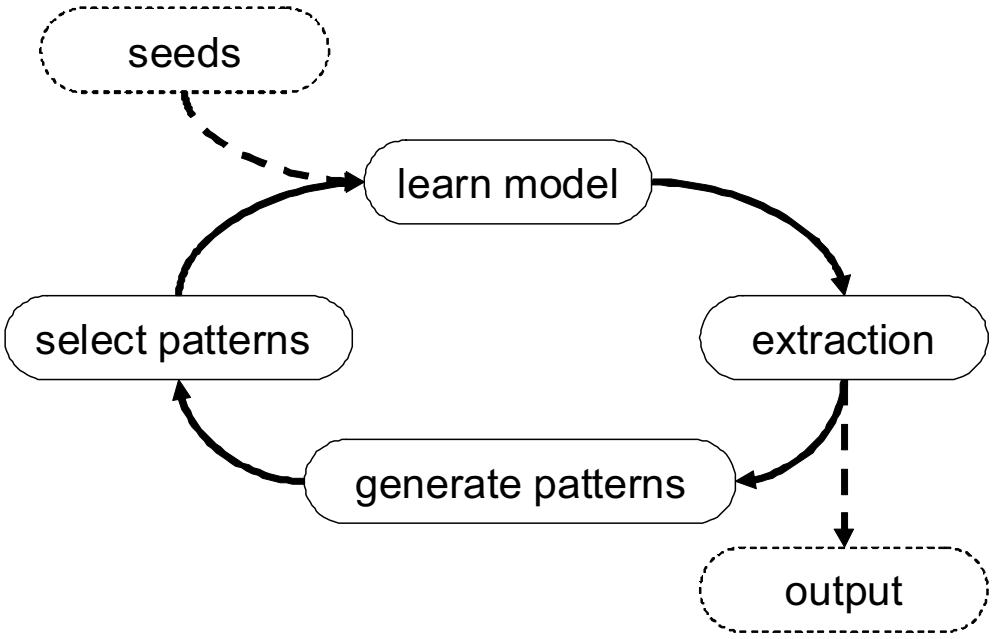


Figure 2.4. The framework of StatSnowball

indicating the continuation of a predicted relation, or O, indicating the token is not believed to be part of an explicit relationship. An illustration is given in Figure 2.3 with the example of the document, ID = 003.

**document id = 003**

Kafka, a writer born in Prague, wrote The Metamorphosis.

Zhu proposed a bootstrapping method using Markov logic networks (Richardson and Domingos, 2006) to extract a relation expression from words located between a given pair (Zhu et al., 2009). Figure 2.4 shows the framework of the method called "StatSnowball". To start the iterative extraction process, the StatSnowball takes the input seeds to learn an extractor. They used the maximum likelihood estimation with word-based features at this step. Then, the learned model is used to extract new relation information from documents. The third step is to generate extraction patterns with the newly identified relation information.

These patterns are used to compose formulae of MLN. Finally, it selects good formulae to add to the probabilistic model and re-train the model. Their method iteratively performs these four steps until no new relation information is identified or no new patterns are generated.

The main contribution of these methods is that they need none or only a few seed examples even they use supervised learning classifiers such as conditional random fields and Markov logic networks.

CHAPTER

# 3

# RELATION DETECTION

## 3.1. Introduction

The relation detection task is detecting semantically-related pairs from named entity pairs that co-occur in a given document. For example, suppose we would like to detect semantically-related pairs from named entity pairs in the document, ID = 002.

**document id = 002**

Minshuto$_1$-no Yamada Ichiro$_2$-wa Yamada Jiro$_3$-no ani$_4$-desu.
( Ichiro Yamada$_2$, the Democratic Party$_1$, is Jiro Yamada$_3$'s brother$_4$. )

There are three named entity pairs in the document, [Ichiro Yamada, Jiro Yamada], [Ichiro Yamada, the Democratic Party] and [the Democratic Party, Jiro Yamada]. Following the definition of semantically-related named entity pairs mentioned in Chapter 1, the pairs [Ichiro Yamada, Jiro Yamada] and [Ichiro Yamada, the Democratic Party] should

be detected as semantically-related ones and [the Democratic Party, Jiro Yamada] should not.

In relation detection task, various supervised learning approaches have been explored to date (Culotta and Sorensen, 2004; Kambhatla, 2004; Zelenko et al., 2003). These approaches automatically learn relation patterns from an annotated corpus. In learning process, they used two kinds of features: syntactic ones and word-based ones. For example, in previous work (Kambhatla, 2004), the path of the given pair in the parse tree and the word n-gram between named entities were used to detect semantically-related named entity pairs. They reported that the syntactic features are especially effective for the relation detection task.

These previous work target only intra-sentential relation detection in which named entity pairs are located in the same sentence, in spite of the fact that about 43.6% of named entity pairs with semantic relations are inter-sentential in Japanese documents. For the inter-sentential task, the prior methods can not detect semantically-related named entity pairs accurately because the key syntactic features are unusable.

To solve the problem, we propose a supervised learning method using contextual features for detecting a semantic relation between a given pair of named entities, which may be located in different sentences.

## 3.2. Contextual features for relation detection

The proposed method uses contextual features based on Salient Referent List (Nariyama, 2002) as well as conventional syntactic and word-based features. These features are organized as a tree structure and are fed into a boosting-based classification algorithm.

Given a named entity pair co-occurred in a document, the proposed method extracts contextual, syntactic and word-based features. Then the method judges whether a given pair is semantically related or not

by classifying a pair using those features. In this section, we describe the underlying idea of contextual features and how contextual features are used for relation detection.

### 3.2.1 Underlying idea of contextual features

When a pair of named entities with a semantic relation appears in different sentences, the antecedent named entity that appear first in a text must be contextually easily referred to in the sentence with the following named entity in the pair. In the following Japanese document, ID = 004, the pair "Ken$_2$" and "Amerika$_8$ (the U.S.)" has a semantic relation, because "Ken$_2$" is contextually referred to in the sentence with "Amerika$_8$" (In fact, the zero pronoun $\phi_i$ refers to "Ken$_2$"). Meanwhile, the pair "Naomi$_5$" and "Amerika$_8$" has no semantic relation, because the sentence with "Amerika$_8$" does not refer to "Naomi$_5$".

**document id = 004**

> asu$_1$, Ken$_2$-wa Osaka$_3$-o otozure$_4$ Naomi$_5$-to au$_6$. sonogo$_7$, ($\phi_i$-ga) Amerika$_8$-ni watari$_9$ Tom$_{10}$-to ryoko$_{11}$ suru.
> (Ken$_2$ is going to visit$_4$ Osaka$_3$ to see$_6$ Naomi$_5$, tomorrow$_1$. Then$_7$, (he$_i$) will go$_9$ to the U.S.$_8$ to travel$_{11}$ with Tom$_{10}$.)

Therefore, it would improve relation detection performance to use whether the antecedent named entity is referred to in the context with the following named entity as features of a given pair of named entities. In this thesis, we use Salient Referent List (Nariyama, 2002) to determine how easily a noun phrase can be referred to in the following context.

### 3.2.2 Salient referent list and preference rules

Centering Theory (Grosz et al., 1983) is a theory about discourse coherence and is based on the idea that each utterance features a topically most salient entity called the center. The main idea of Centering Theory is that certain entities mentioned in an utterance are more central

in discourse than others and this imposes certain constraints on the use of referring expressions and in particular on the use of pronouns. As an extension of this theory, Nariyama (Nariyama, 2002) proposed an algorithm of zero anaphora resolution, including salient referent list and preference rules. The salient referent list can deal with entities in all of the preceding utterances, whereas the original Centering Theory does only account for the entities in the immediately preceding utterance. Furthermore, if there are more than one zero pronouns in the target sentence, her algorithm identifies an antecedent among each entity in the salient referent list for a given zero-pronoun according to the following preference rules.

- Topicalized Subject (wa) $>$ Subject (ga) $>$ Indirect Object (ni) $>$ Object (o) $>$ Others

The preference rules are based on natures that topicalized subject has a tendency to be omitted and to be marked by particle "wa" in Japanese. Salient referent list has stacks, last-in first-out structure, for each element, topicalized subject, subject, indirect object, object and others, in the preference rules.

To identify the antecedent of a given pronoun, from the beginning of the text until the pronoun appears, noun phrases are pushed to the corresponding stack based on their particles. Then the stacked information is sorted by the preference rules and stacks.

In the example described in Section 3.2.1, noun phrases, "asu$_1$", "Ken$_2$", "Osaka$_3$" and "Naomi$_5$", which are in the previous context of the zero pronoun $\phi_i$, are stacked and then the information shown in Figure 3.1 is acquired. The stacked information is sorted by the preference rules and stacks then the order, 1: "Ken$_2$", 2: "Osaka$_3$", 3: "Naomi$_5$", 4: "asu$_1$", is assigned. In this way, using salient referent list would show that the antecedent of the zero pronoun $\phi_i$ is "Ken$_2$".

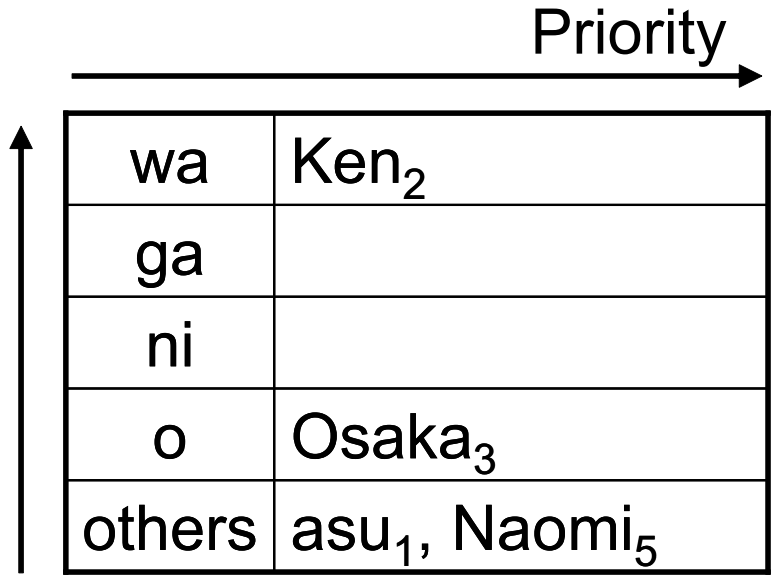


Figure 3.1. Stacked Information on Salient Referent List

### 3.2.3 Applying Salient Referent List to Relation Detection

To judge whether a given pair of named entities is semantically related or not, we use Salient Referent List to determine how easily the antecedent named entity in the pair can be referred to in the context with the following named entity. Note that we do not explicitly execute anaphora resolutions here.

#### Top Instance in Salient Referent List

To apply Salient Referent List to relation detection task, we slightly change the condition of stacking process in the algorithm described in Section 3.2.2. To judge whether a given pair of named entities is semantically related or not, from the beginning of the text until **the following named entity in the pair** appears, noun phrases are pushed to the corresponding stack based on their particles. Then the stacked information is sorted by the preference rules and stacks. Our proposed

method use whether the antecedent named entity in the pair is the top instance in the sorted order as a contextual feature. If the top instance in the sorted order is identical to the antecedent named entity, we suppose the antecedent named entity is easily referred to in the context with the following named entity. When the top instance in the sorted order is identical to the antecedent named entity, the value of contextual feature, called "SRL-T" (Salient Referent List Top) in this thesis, is "1".

When the pair of named entities, "Ken$_2$" and "Amerika$_8$", is given in the example described in Section 3.2.1, the noun phrases, "asu$_1$", "Ken$_2$", "Osaka$_3$" and "Naomi$_5$", which are in the previous context of the following named entity "Amerika$_8$", are pushed to the corresponding stack based on their particles and then the information shown in Figure 3.1 is acquired.

Then the stacked information is sorted by the preference rules and stacks then the order, 1: "Ken$_2$", 2: "Osaka$_3$", 3: "Naomi$_5$", 4: "asu$_1$", is assigned. Here, because the top instance in the sorted order is identical to the antecedent named entity, we suppose "Ken$_2$" is easily referred to in the context with "Amerika$_8$", and the value of contextual feature "SRL-T" becomes "1". By contrast, when the pair of named entities, "Naomi$_5$" and "Amerika$_8$", is given in the same example, because the top instance in the sorted order is not identical to the antecedent named entity, we suppose "Naomi$_5$" is not referred to in the context with "Amerika$_8$", and the value of contextual feature "SRL-T" becomes "0".

Using the top instance in Salient Referent List as contextual features, we expect that it is possible to judge accurately whether a given pair is semantically related.

#### Structure of Salient Referent List

Because the preference rules described in Section 3.2.2 are based on natures that topicalized subject has a tendency to be omitted in Japanese, the top instance in the sorted order by the preference rules

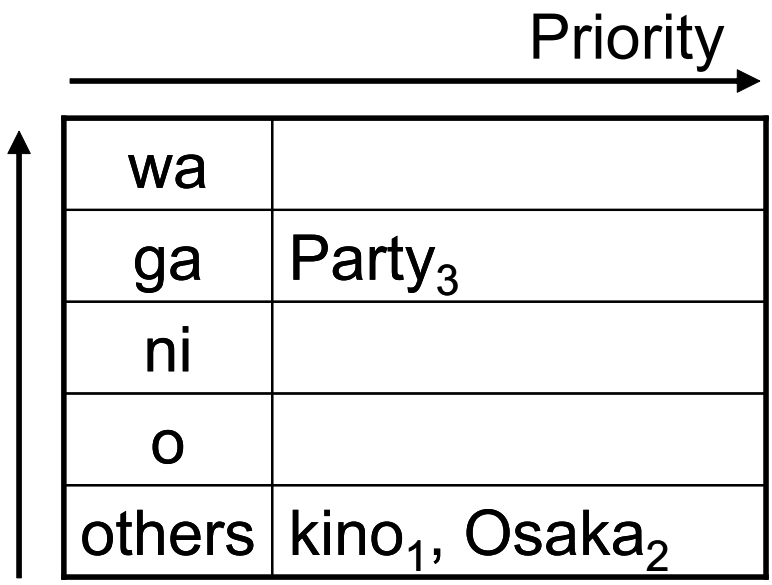


Figure 3.2. Stacked Information on Salient Referent List for the pair, "Osaka$_2$" and "Ken$_5$" in the document, id = 005.

must be a topicalized subject, such as person and organization. Therefore, when the antecedent named entity in the given pair is not a topicalized subject, such as location, using the top instance in Salient Referent List suppose the antecedent named entity is not referred to in the context with the following named entity.

For example, when the pair of named entities, "Osaka$_2$" and "Ken$_5$", is given in the document ID = 005, the noun phrases, "kino$_1$", "Osaka$_2$" and "Party$_3$", which are in the previous context of the following named entity "Ken$_5$" are pushed to the corresponding stack based on their particles and then the information shown in Figure 3.2 is acquired.

**document id = 005**

kino$_1$, Osaka$_2$-de party$_3$-ga atta$_4$. Ken$_5$-ga sanka$_6$-shita.
(There was$_4$ a party$_3$ in Osaka$_2$, yesterday$_1$. Ken$_5$ participated$_6$ it.)

The stacked information is sorted by the preference rules and stacks then the order, 1: "Party$_3$", 2: "Osaka$_2$", 3: "kino$_1$", is assigned. Here,

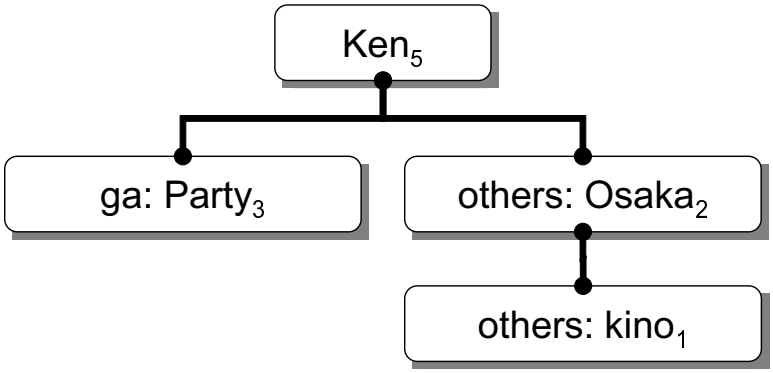


Figure 3.3. Structure of Salient Referent List for the pair, "Osaka$_2$" and "Ken$_5$" in the document, id = 005.

because the top instance in the sorted order is not identical to the antecedent named entity, "Osaka$_2$" is supposed not referred to in the context with "Ken$_5$", and the value of contextual feature "SRL-T" becomes "0".

As far as relation detection task is concerned, we would like to judge the above pair is semantically related. So, for the case that the antecedent named entity in the given pair is not a topicalized subject, such as location, we use the structure of Salient Referent List without the preference rules as contextual features.

A method to acquire the structure of Salient Referent List is follows. First, from among the given entities, we choose the one that appears last in the documents as the root of the tree. We then use the following rules to append noun phrases from the chosen one to the beginning of the document, to the tree according to case markers, "wa" (Topicalized Subject), "ga" (Subject), "ni" (Indirect Object), "o" (Object), and "others". If there are nodes of the same case marker already in the tree, the noun phrase is appended as a child of their leaf node. In other cases, the noun phrase is appended as a child of the root node. For example, we create the structure of SRL shown in Figure 3.3 for the given entity pair, $X$ = "Osaka$_2$" and $Y$ = "Ken$_5$", in the document, id = 005. First, from $X$ = "Osaka$_2$", $Y$ = "Ken$_5$", we choose $Y$ = "Ken$_5$" that appears last in the

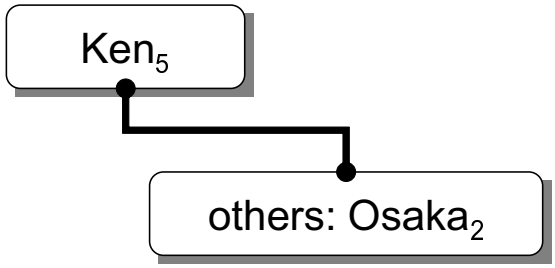


Figure 3.4. The minimal tree that consists of given named entities, "Osaka$_2$" and "Ken$_5$" in the document, id = 005.

document as the root node. Next, from "Ken$_5$" to the beginning of the document, "Party$_3$" is appended with case marker "ga" as a child of the root node, "Osaka$_2$" is appended with case marker "others" as a child of the root node, and "kino$_1$" is appended with case marker "others" as a child of "Osaka$_2$". In Figure 3.3, the depth of the tree represents the referential degree of phrases for each case marker in the context in which the root phrase appears in a document, for example, for case marker "others", phrase "Osaka$_2$" is more referential than "kino$_1$" in the context of the appearance of "Ken$_5$".

To use the structure of SRL as contextual features for detecting semantically related pairs, we make a minimal tree that consists of given named entities, called "SRL-S" (Salient Referent List Structure). Figure 3.4 shows the contextual features 'SRL-S".

Using the structure of Salient Referent List as contextual features, we expect that it is possible to judge accurately whether a given pair is semantically related, even for the case that the antecedent named entity in the given pair is not a topicalized subject, such as location.

### 3.2.4 Classification Algorithm

In this thesis, we use structure-based learning algorithms that have good reported performance among several learning algorithms that use structural information, such as Tree kernel (Collins and Duffy, 2002),

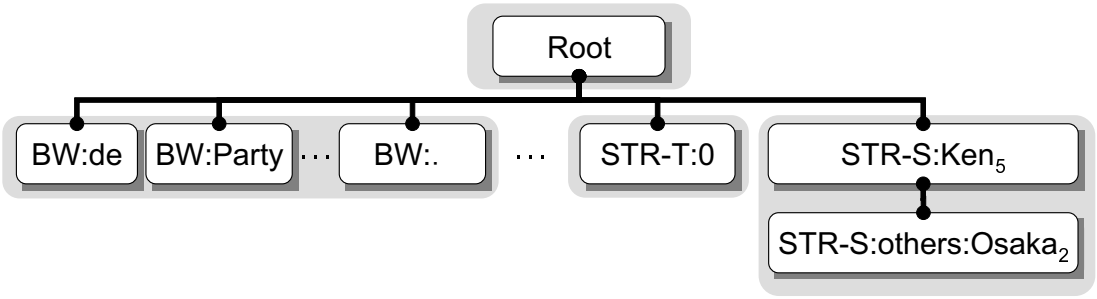


Figure 3.5. Features organized as a tree

HDAG kernel (Suzuki et al., 2003), or boosting-based algorithm (Kudo and Matsumoto, 2004). The experiments tested Kudo and Matsumoto's boosting-based algorithm using sub-trees as features; it has comparatively short learning times and has been implemented as a BACT system[1].

In out proposed method, the contextual features and conventional features, syntactic and word-based features, are organized as a tree structure. These features are organized as a tree structure and are fed into a boosting-based classification algorithm. Here, contextual feature SRL-T and word-based features are not structural features, so we suppose these features are structure features consisting of one node. To organize all features as a tree, we first prepare the root node marked "Root". Then, we put each structure feature as a child of the root node in the tree with the label indicating feature type, such as SRL-T, SRL-S or etc. For example, Figure 3.5 shows that the features organized as a tree when the pair of named entities, "$\text{Osaka}_2$" and "$\text{Ken}_5$", is given in the document, ID = 005.

Using the tree structure, given a set of training examples, each of which is represented as a tree labeling whether the named entity pair is semantically related or not, the BACT system learns a set of rules that is effective in classification. In the experiments, we used the BACT system with option $L$ = 5, which restricts the maximun size of trees.

[1] http://chasen.org/~taku/software/bact/

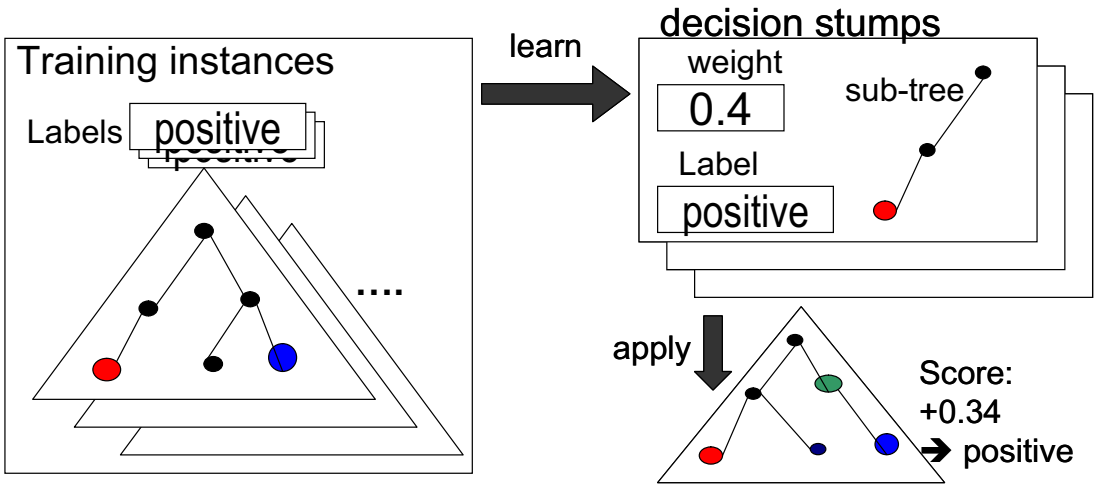


Figure 3.6. Overview of the BACT system

Then, given a test instance, the BACT system classifies it using the set of learned rules (Figure 3.6).

## 3.3. Experiments

We conducted experiments using texts from Japanese newspaper articles and weblog texts to test the proposed method against both intra- and inter-sentential tasks. In the experiments, we compared the following methods:

**WD:** Detecting named entity pairs when pairs appeared within $n$ words in documents.

**DEP:** Supervised learning method using syntactic and word-based features, the path of the pairs of named entities in dependency tree and the word n-gram between pairs (Kambhatla, 2004).

**DEP+SRL-T:** Supervised learning method using features of syntactic, word-based and top instance in Salient Referent List.

Table 3.1. The inter-annotator agreement, $\kappa$.

| | $\kappa$ | person $A$ | person $B$ | Total |
|---|---|---|---|---|
| person ↔ person | 0.802 | 8,993 | 8,026 | 127,186 |
| person ↔ organization | 0.882 | 3,228 | 3,116 | 99,084 |
| person ↔ location | 0.823 | 4,563 | 4,026 | 111,121 |
| organization ↔ organization | 0.851 | 1,448 | 1,334 | 45,736 |
| organization ↔ location | 0.800 | 2,220 | 2,002 | 73,919 |
| location ↔ location | 0.841 | 3,877 | 3,759 | 80,365 |
| Total | 0.827 | 24,329 | 22,263 | 537,411 |

Table 3.2. The inter-annotator agreement, *Precision* and *Recall*.

| | *Precision* [%] | *Recall* [%] |
|---|---|---|
| person ↔ person | 86.4 ( 6,942 / 8,026 ) | 77.1 ( 6,942 / 8,993 ) |
| person ↔ organization | 90.1 ( 2,810 / 3,116 ) | 87.0 ( 2,810 / 3,228 ) |
| person ↔ location | 88.5 ( 3,567 / 4,026 ) | 78.1 ( 3,567 / 4,563 ) |
| organization ↔ organization | 89.2 ( 1,191 / 1,334 ) | 82.2 ( 1,191 / 1,448 ) |
| organization ↔ location | 84.9 ( 1,701 / 2,002 ) | 76.6 ( 1,701 / 2,220 ) |
| location ↔ location | 86.1 ( 3,240 / 3,759 ) | 83.5 ( 3,240 / 3,877 ) |
| Total | 87.3 ( 19,451 / 22,263 ) | 79.9 ( 19,451 / 24,329 ) |

**DEP+SRL-S:** Supervised learning method using features of syntactic, word-based and structure of Salient Referent List.

**DEP+SRL-T+SRL-S:** Supervised learning method using features of syntactic, word-based, top instance in Salient Referent List and structure of Salient Referent List.

### 3.3.1 Setting

We took 6,200 documents from Japanese newspapers and weblogs dated from January 1, 2004 to June 30, 2006, which were obtained by a web crawler, and manually annotated the semantically-related pairs of named entities for experimental purposes. The named entity

Table 3.3. Percentage of semantically-related pairs in annotated data.

| | *intra − sentential* [%] | *inter − sentential* [%] |
|---|---|---|
| person ↔ person | 38.4 ( 3,924 / 10,213 ) | 4.3 ( 5,069 / 116,973 ) |
| person ↔ organization | 30.5 ( 2,035 / 6,683 ) | 1.3 ( 1,193 / 92,401 ) |
| person ↔ location | 35.3 ( 2,404 / 6,805 ) | 2.1 ( 2,159 / 104,316 ) |
| organization ↔ organization | 26.4 ( 998 / 3,780 ) | 1.1 ( 450 / 41,956 ) |
| organization ↔ location | 26.4 ( 1,402 / 5,317 ) | 1.2 ( 818 / 68,602 ) |
| location ↔ location | 38.0 ( 2,955 / 7,774 ) | 1.3 ( 922 / 72,591 ) |
| Total | 33.8 ( 13,718 / 40,572 ) | 2.1 ( 10,611 / 496,839 ) |

pairs targeted in this data were person ↔ person, person ↔ organization, person ↔ location, organization ↔ organization, organization ↔ location, and location ↔ location. We gave annotators the definition of semantically-related pairs described in Chapter 1 and the instruction, "Select any named entity pair that follows the definition from named entity pairs that co-occur in a given document."

To investigate inter-annotator agreement, two people annotated a total of 537,411 pairs in 6,200 documents. Tables 3.1 and 3.2 show the result. The agreement of the two annotators was $\kappa$ = 0.827, person $A$ selected 24,329 pairs and person $B$ selected 22,263 pairs out of 537,411 pairs. We also calculated precision and recall by assuming that the annotated data produced by person $A$ was the answer, the data produced by person $B$ the system; precision was 87.3% and recall was 79.9%. The inter-annotator agreement is high as seen above, so we use the annotated data produced by a single person in the experiments.

Table 3.3 shows details of the annotated data produced by a single person for a total of 537,411 named entity pairs, 13,718 semantically-related pairs appeared in the same sentence (intra-sentential), while 10,611 semantically-related pairs appeared in different sentences (inter-sentential). We used the semantically-related pairs as positive examples and the rest pairs as negative examples. In the intra-sentential experiment, 40,572 pairs were given, but only 13,718 of them are semantically-related. In contrast, in the inter-sentential experiment,

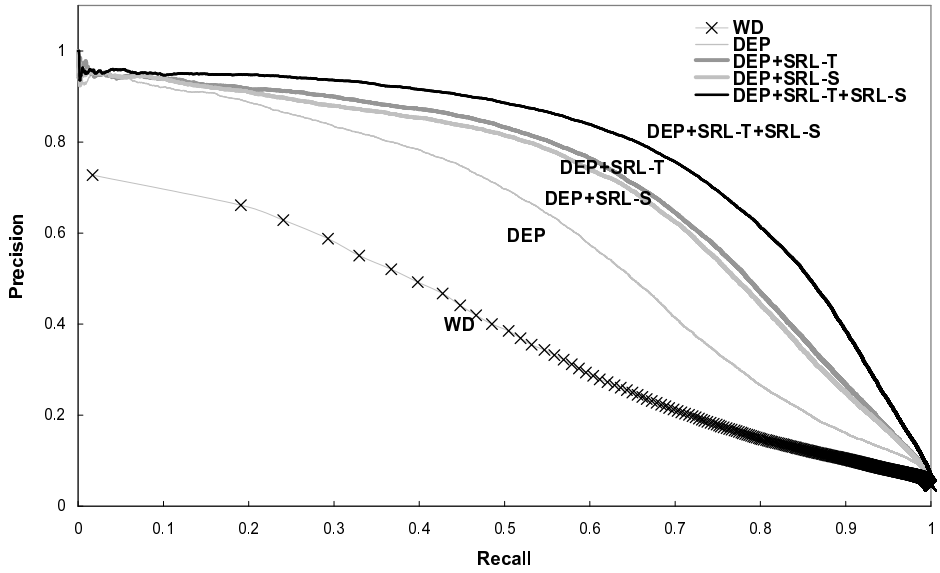


Figure 3.7. Recall-precision curve

496,839 pairs were given, and only 10,611 of them are semantically-related.

We conducted five-fold cross-validation over 40,572 pairs in intra-sentential or 496,839 pairs in inter-sentential so that sets of pairs from a single document were not divided into the training and test sets. In the experiments, all features were automatically acquired using a Japanese POS tagger (Fuchi and Takagi, 1998) and dependency parser (Imamura et al., 2007).

### 3.3.2 Results and Discussion

Figure 3.7 shows the performance of each method for relation detection. *Precision* is defined as the percentage of correct detected pairs out of all those detected. *Recall* is the percentage of correct detected pairs out of the manual annotation results. We plotted recall-precision curves altering threshold parameters; value of discriminant function in the classifier. In "WD" method, we plotted recall-precision curves

Table 3.4. Results of intra-sentential task, *Precision* and *Recall*.

| | *Precision* [%] | *Recall* [%] |
|---|---|---|
| WD10 | 55.0 ( 10,416 / 18,934 ) | 75.9 ( 10,416 / 13,718 ) |
| DEP | 72.0 ( 9,517 / 13,227 ) | 69.4 ( 9,517 / 13,718 ) |
| DEP+SRL-T | 75.3 ( 10,011 / 13,299 ) | 73.0 ( 10,011 / 13,718 ) |
| DEP+SRL-S | 74.5 ( 9,870 / 13,242 ) | 71.9 ( 9,870 / 13,718 ) |
| DEP+SRL-T+SRL-S | 79.3 ( 10,615 / 13,392 ) | 77.4 ( 10,615 / 13,718 ) |

Table 3.5. Results of inter-sentential task, *Precision* and *Recall*.

| | *Precision* [%] | *Recall* [%] |
|---|---|---|
| WD10 | 11.5 ( 938 / 8,148 ) | 8.8 ( 938 / 10,611 ) |
| DEP | 60.9 ( 2,842 / 4,666 ) | 26.8 ( 2,842 / 10,611 ) |
| DEP+SRL-T | 78.4 ( 4,649 / 5,933 ) | 43.8 ( 4,649 / 10,611 ) |
| DEP+SRL-S | 75.9 ( 4,475 / 5,899 ) | 42.2 ( 4,475 / 10,611 ) |
| DEP+SRL-T+SRL-S | 82.9 ( 5,210 / 6,286 ) | 49.1 ( 5,210 / 10,611 ) |

altering word distance $n$.

Comparing "DEP" to "DEP+SRL-T+SRL-S" for relation detection task indicates that the proposed method based on contextual features improved performance, *Precision* by 11.3 points and *Recall* by 14.2 points, when the threshold parameter was 0; "DEP": *Precision* = 69.1 and *Recall* = 50.8, "DEP+SRL-T+SRL-S": *Precision* = 80.4 and *Recall* = 65.0. This result supports our idea that it is useful to accumulate contextual features for relation detection between named entities.

To investigate the effectiveness of proposed method for inter-sentential task, tables 3.4 and 3.5 show the performance of each method for intra-/inter-sentential relation detection tasks. Comparing "DEP" to "DEP+SRL-T+SRL-S" for inter-sentential task indicates that the proposed method based on contextual features improved performance, *Precision* by 22.0 points and *Recall* by 22.3 points.

At last, to investigate the effectiveness of proposed method for each targeted pairs, person $\leftrightarrow$ person, person $\leftrightarrow$ organization, person $\leftrightarrow$ location, organization $\leftrightarrow$ organization, organization $\leftrightarrow$ location, and

Table 3.6. Results of target pairs, *Precision* and *Recall*.

| | *Precision* [%] | *Recall* [%] |
|---|---|---|
| person ↔ person | | |
| WD10 | 49.5 ( 2,961 / 5,983 ) | 32.9 ( 2,961 / 8,993 ) |
| DEP | 77.2 ( 5,251 / 6,799 ) | 58.4 ( 5,251 / 8,993 ) |
| DEP+SRL-T | 84.9 ( 6,841 / 8,054 ) | 76.1 ( 6,841 / 8,993 ) |
| DEP+SRL-S | 84.2 ( 6,623 / 7,870 ) | 73.6 ( 6,623 / 8,993 ) |
| DEP+SRL-T+SRL-S | 87.2 ( 7,124 / 8,169 ) | 79.2 ( 7,124 / 8,993 ) |
| person ↔ organization | | |
| WD10 | 48.6 ( 1,802 / 3,709 ) | 55.8 ( 1,802 / 3,228 ) |
| DEP | 66.8 ( 1,603 / 2,399 ) | 49.7 ( 1,603 / 3,228 ) |
| DEP+SRL-T | 72.5 ( 1,695 / 2,338 ) | 52.5 ( 1,695 / 3,228 ) |
| DEP+SRL-S | 71.1 ( 1,705 / 2,398 ) | 52.8 ( 1,705 / 3,228 ) |
| DEP+SRL-T+SRL-S | 78.1 ( 1,893 / 2,425 ) | 58.6 ( 1,893 / 3,228 ) |
| person ↔ location | | |
| WD10 | 42.0 ( 1,904 / 4,535 ) | 41.7 ( 1,904 / 4,563 ) |
| DEP | 61.9 ( 1,865 / 3,014 ) | 40.9 ( 1,865 / 4,563 ) |
| DEP+SRL-T | 68.6 ( 2,124 / 3,094 ) | 46.5 ( 2,124 / 4,563 ) |
| DEP+SRL-S | 68.2 ( 2,125 / 3,116 ) | 46.6 ( 2,125 / 4,563 ) |
| DEP+SRL-T+SRL-S | 75.4 ( 2,401 / 3,183 ) | 52.6 ( 2,401 / 4,563 ) |
| organization ↔ organization | | |
| WD10 | 35.5 ( 851 / 2,399 ) | 58.8 ( 851 / 1,448 ) |
| DEP | 57.2 ( 591 / 1,034 ) | 40.8 ( 591 / 1,448 ) |
| DEP+SRL-T | 68.1 ( 747 / 1,097 ) | 51.6 ( 747 / 1,448 ) |
| DEP+SRL-S | 63.9 ( 697 / 1,090 ) | 48.1 ( 697 / 1,448 ) |
| DEP+SRL-T+SRL-S | 73.8 ( 866 / 1,174 ) | 59.8 ( 866 / 1,448 ) |
| organization ↔ location | | |
| WD10 | 32.7 ( 1,195 / 3,650 ) | 53.8 ( 1,195 / 2,220 ) |
| DEP | 56.3 ( 821 / 1,459 ) | 37.0 ( 821 / 2,220 ) |
| DEP+SRL-T | 64.3 ( 938 / 1,458 ) | 42.3 ( 938 / 2,220 ) |
| DEP+SRL-S | 61.0 ( 921 / 1,509 ) | 41.5 ( 921 / 2,220 ) |
| DEP+SRL-T+SRL-S | 71.8 ( 1,071 / 1,492 ) | 48.2 ( 1,071 / 2,220 ) |
| location ↔ location | | |
| WD10 | 38.8 ( 2,641 / 6,806 ) | 68.1 ( 2,641 / 3,877 ) |
| DEP | 69.9 ( 2,228 / 3,188 ) | 57.5 ( 2,228 / 3,877 ) |
| DEP+SRL-T | 72.5 ( 2,315 / 3,191 ) | 59.7 ( 2,315 / 3,877 ) |
| DEP+SRL-S | 72.0 ( 2,274 / 3,158 ) | 58.7 ( 2,274 / 3,877 ) |
| DEP+SRL-T+SRL-S | 76.4 ( 2,470 / 3,235 ) | 63.7 ( 2,470 / 3,877 ) |

location ↔ location, table 3.6 shows the performance of each method. The proposed method improved performances of all named entity pairs and was most effective for organization ↔ organization pair, $Precision$ by 16.6 points and $Recall$ by 19.0 points.

### 3.3.3 Error Analysis

The above experiments showed that our proposed method based on contextual features is effective in relation detection between named entities. However, the experiments identified some challenging problems that need to be overcome to improve the method further. Here, we elucidate the remaining problems by analyzing the two main types of errors, which cover over 80% of the errors.

#### Definite anaphora

The proposed method did not detect semantically-related pairs liked by definite nouns phrases, such as “shusho (the prime minister)” or “shacho (the president)”. In English, definite nouns phrases are nouns phrases with definite article. However, because there are no articles in Japanese, we need to judge whether definite nouns phrase or not.

#### Topic segment

The proposed method detected un-semantically-related pairs appeared in different topics in a document, such as “SPORT” and “ELECTION”. When a pair appeared in different topics in a document, the pair must not be semantically-related in almost all cases. Therefore the method needs to judge topic segments in a document.

## 3.4. Conclusion

The relation detection task is detecting semantically-related pairs from named entity pairs that co-occur in a given document. The previous

work used two kinds of features: syntactic ones and word-based ones, for example, the path of the given pair in the parse tree and the word n-gram between named entities. They target only intra-sentential relation detection in which named entity pairs are located in the same sentence, in spite of the fact that about 43.6% of named entity pairs with semantic relations are inter-sentential in Japanese documents. Our proposed method is a supervised learning method using contextual features for detecting a semantic relation between a given pair of named entities, which may be located in different sentences. Our experiments demonstrated that the method improves $Precision$ by 11.3 points and $Recall$ by 14.2 points and thus helps to detect semantically-related pairs between named entities.

CHAPTER

# 4

# RELATION EXPRESSION RECOGNITION

## 4.1. Introduction

The relation expression recognition task is recognizing a relation expression that demonstrates the explicit relation between the semantically related pair co-occur in the document. For example, suppose we would like to recognize a relation expression between a semantically related pair, [Ichiro Yamada$_2$, Jiro Yamada$_3$], in the following document.

**document id = 002**

Minshuto$_1$-no Yamada Ichiro$_2$-wa Yamada Jiro$_3$-no ani$_4$-desu.
( Ichiro Yamada$_2$, the Democratic Party$_1$, is Jiro Yamada$_3$'s brother$_4$. )

There are three candidates in this case, "the Democratic Party$_1$", "brother$_4$" and "None". "None" means that there is no relation ex-

pression in the document. From these candidates, "brother" should be recognized as a relation expression between the given pair, [Ichiro Yamada$_2$, Jiro Yamada$_3$].

In previous work, Banko and Etzioni (2008) proposed a supervised learning method using conditional random fields to recognize a relation expression from words located between a given pair. Zhu et al. (2009) also proposed a bootstrapping method using Markov logic networks to recognize a relation expression from words located between a given pair. They designed the task as selecting relation expressions only from the words between the given entities in the document. It is impractical to apply previous methods to Japanese texts, because in our annotated data described in Section 4.3.1 only 26% of relation expressions appear between the pair. On the other hand, as far as English documents are concerned, 86% of the relation expressions appear between the pair (Banko and Etzioni, 2008).

To recognize relation expressions in Japanese documents, we designed the task as selecting, from the entire document, a phrase (such as noun phrase or verb phrase) that does not cross a bunsetsu boundary and includes the relation expression that connects the given pair. If a relation expression belongs to more than one bunsetsu, the phrase that belongs to the last bunsetsu of the relation expression must be selected. Although this condition creates the problem that the recognized relation expression could be different from ones asserted in the document, we consider that this problem will be solved by rule-based post-processing using recognized relation expressions as clues. The task also includes judging which named entity is the subject of the relation expressions.

When recognizing a relation expression for a given named entity pair in a given document we must consider two cases. One is that the named entities appear in the same sentence, and the other is that they appear in different sentences. In the former, the dependency structure of the sentence containing the pair can be used as an informative feature. In the latter, the discourse structure of the document, the

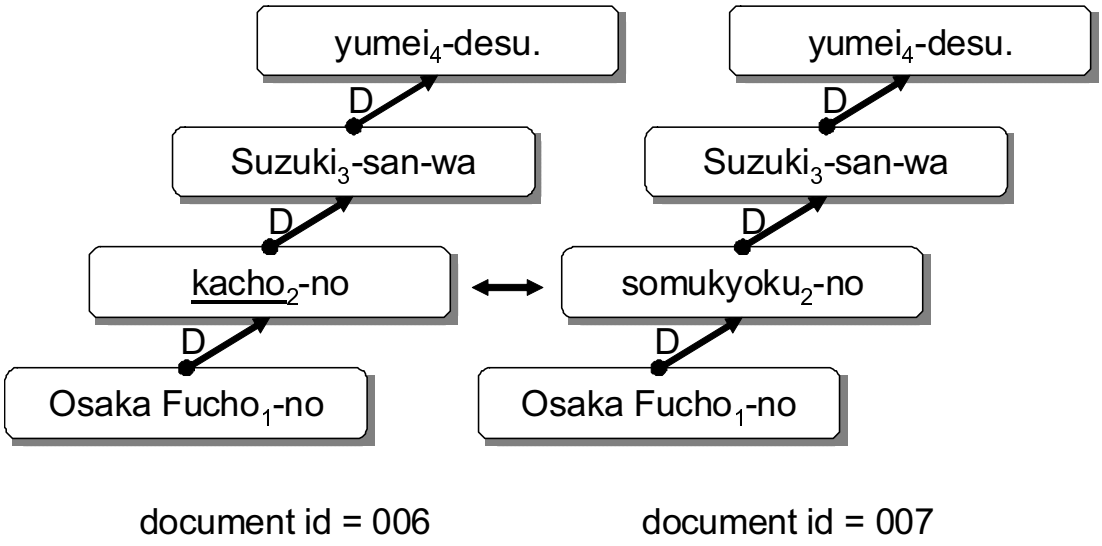


Figure 4.1. Examples of the same dependency structure, document id = 006 and 007

structure across sentences, can be used even though it is less informative than the dependency structure. We assume that knowledge can be acquired from the former case and then reused in the latter case as informative features. In this thesis, we focus on the case in which the named entities appear in the same sentence.

In a preliminary observation, we found that word-based features, even syntactic ones, were rather infrequent in a number of samples. For instance, the following two Japanese documents have the same dependency structure, see Figure 4.1, and the same semantically related pair, [Osaka Fucho$_1$, Suzuki$_3$], but document ID = 006 contains a relation expression (e.g. kacho$_2$) while document ID = 007 does not.

**document id = 006**

Osaka Fucho$_1$-no kacho$_2$-no Suzuki$_3$-san-wa yumei$_4$-desu.
(Mr. Suzuki$_3$, a manager$_2$ of Osaka Prefectural Government$_1$, is famous$_4$.)

**document id = 007**

Osaka Fucho$_1$-no somukyoku$_2$-no Suzuki$_3$-san-wa yumei$_4$-desu.

(Mr. Suzuki$_3$, administration office$_2$ in Osaka Prefectural Government$_1$, is famous$_4$.s)

To solve the above problem, we propose a supervised learning method using two kinds of external information about candidates acquired from large text corpora automatically. One is lexical information with selected nouns that indicate relations. The other is relation predicting model which predicts present relations between named entities on the basis of past relations of the pair.

## 4.2. Recognizing relation expressions

To recognize relation expressions for given pairs from Japanese documents, we need to select, from among all noun and verb phrases that do not cross bunsetsu in a given text, the phrase that includes the relation expression. Candidate phrases can lie in the same sentence as the given pair (intra-sentential), or in another sentence (inter-sentential). For example, the relation expression "taiketsu$_2$" between the pair [Tom$_5$, Ken$_6$] in the following document is inter-sentential.

**document id = 008**

Chumoku$_1$-no taiketsu$_2$-ga mamonaku$_3$ hajimaru$_4$. Tom$_5$-to Ken$_6$-niyoru yume$_7$-no kikaku$_8$.
(The showcase$_1$ match$_2$ will start$_4$ soon$_3$. (It is) a dream$_7$ event$_8$ by Tom$_5$ and Ken$_6$.)

According to our annotated data (Japanese documents), 53.3% of the semantically-related named entity pairs are intra-sentential and 11.9% are inter-sentential (See Section 4.3.1 for details). Thus, we first attempt to select an intra-sentential phrase; if no such phrase is found, we select inter-sentential phrases. As far as we know, our work represents the first attempt to recognize inter-sentential relation expressions in Japanese documents.

In this thesis, we propose a supervised learning method that uses two novel features based on inherent clues of candidate words and a

relation predicting model as well as structural features. These features are organized into a tree structure and are fed into a boosting-based classification algorithm (Kudo and Matsumoto, 2004). In the classification, to judge which named entity is the subject, $X$, of relation expressions, we classify a tree structure by two models, one model treats the first named entity of the pair as the subject, and the other model treats the second named entity as the subject. The highest-scoring phrase is then selected if the score exceeds a given threshold. Finally, the word sequence in the selected phrase excluding functional words is output as the relation expression of the given entity pair.

The method consists of four parts: preprocessing (POS tagging and dependency parsing), feature extraction, classification, and selection. In this section, we describe the idea behind using our two novel features and how they are implemented so as to recognize the relation expressions of given pairs. We start by describing our proposed method's conventional features.

### 4.2.1 Conventional Features

#### Dependency Structure

To recognize the intra-sentential relation expression for a given pair, we assume that there is a domain independent extraction pattern of dependency structure that consists of given entities and their relation expression. For example, there is an extraction pattern that sets a relation expression, "ani$_4$", as the common parent bunsetsu of the given pair, [Yamada Ichiro$_2$, Yamada Jiro$_3$], for the Japanese document, id = 002. Figure 4.2 (a) shows the dependency structure of the sentence.

To discover the extraction patterns, for each candidate, we make a minimal tree that consists of given entities and the candidate where each bunsetsu is represented by the node "Bunsetsu" having child nodes with case marker "CM", dependency type "DT", an entity class, and the string "STR" and POS of the candidate (See Figure 4.2 (b)).

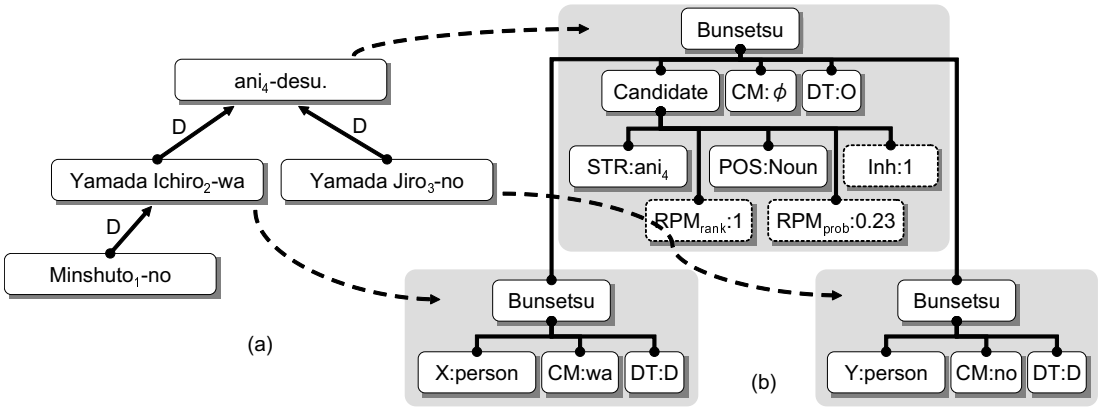


Figure 4.2. (a) The dependency structure and (b) the intra-sentential feature tree of $X$="Yamada Ichiro$_2$", $Y$="Yamada Jiro$_3$", and Candidate="ani$_4$" in the document, id = 002.

**Discourse Structure**

Suppose that the relation expression between named entity pair lies in a different sentence from the pair, we must address two cases of relation expressions. One case is that the relation expressions lie in a sentence that precedes the one holding the entity pair, and the other case is that the relation expressions lie in a following sentence. In the latter case, to extract relation expressions for a given pair, we need to use information to show the given pair is referential from the context in which the relation expressions appear. This information is often used in the research of predicate argument analysis. For example, the Salient Referent List (**SRL**) (Nariyama, 2002) was proposed to identify the antecedent of (zero) pronouns, and shows the discourse structure of texts in conformance to the centering theory (Grosz et al., 1983). In the former case, there are examples in which the sentence that holds the entity pair is missing the relation expression due to ellipsis or coreferencing. For example, in the document, ID = 008, the relation expressions of the given pair are missing from the sentence holding the named entities, because noun "kikaku$_8$" is coreferred from relation expression "taiketsu$_2$" in a previous sentence.

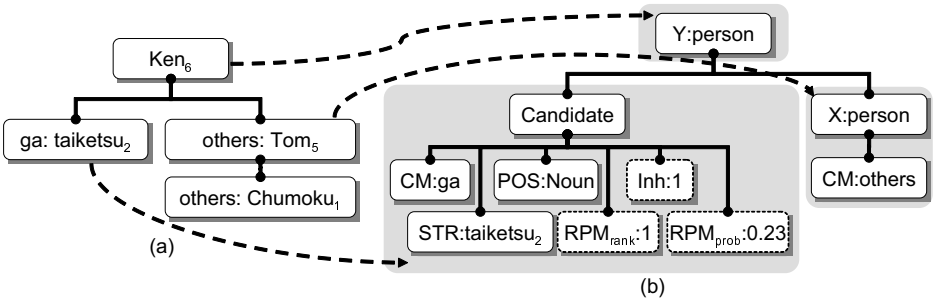


Figure 4.3. (a) The salient referent list and (b) the inter-sentential feature tree of $X$="Tom$_5$", $Y$="Ken$_6$", and candidate="taiketsu$_2$" in the document, id = 008.

To recognize the relation expressions for a given pair, we need to use information that can show that the relation expression is referential from the context of the named entities. We consider that the same information as the latter case, SRL, can also be used in the former case.

Given a document and a target position, SRL identifies which phrases are referential in the context of the target position. For example, given a document, id = 008, and target position "Ken$_6$", SRL outputs the discourse structure shown in Figure 4.3 (a). In Figure 4.3 (a), the depth of the tree represents the referential degree of phrases for each case marker in the context in which the root phrase appears in a document, for example, for case marker "others", phrase "Tom$_5$" is more referential than "Chumoku$_1$" in the context of the appearance of "Ken$_6$".

Using this discourse structure and the following preference rule of case marker, phrases are sorted in order of referential degree, 1:"taiketsu$_2$", 2:"Tom$_5$", 3:"Chumoku$_1$".

- Topicalized Subject (wa) $>$ Subject (ga) $>$ Indirect Object (ni) $>$ Object (o) $>$ Others

To recognize the inter-sentential relation expression for a given pair, we assume that there is a domain independent extraction pattern of

discourse structure that consists of given entities and their relation expression. For example, in the context that one entity, $Y$ = "Ken$_6$", is appeared in the document id = 008, the other entity, $X$ = "Tom$_5$", and the relation expression, $R$ = "taiketsu$_2$" are referential in discourse.

SRL is an empirical sorting rule proposed to identify the antecedent of (zero) pronouns (Nariyama, 2002), and Hirano et al. (2007) proposed a way of applying SRL to relation detection. In this work, we adopt their approach and use SRL to recognize inter-sentential relation expressions.

We apply SRL to each candidate as follows. First, from the given entities and the candidate, we choose the one that appears last in the documents as the root of the tree. We then use the following rules to append noun phrases from the chosen one to the beginning of the document (not across bunsetsu boundaries), to the tree according to case markers, "wa" (Topicalized Subject), "ga" (Subject), "ni" (Indirect Object), "o" (Object), and "others". If there are nodes of the same case marker already in the tree, the noun phrase is appended as a child of their leaf node. In other cases, the noun phrase is appended as a child of the root node. For example, we create the SRL shown in Figure 4.3 (a) for the given entity pair, $X$ = "Tom$_5$" and $Y$ = "Ken$_6$", and the candidate, "taiketsu$_2$", in the document, id = 008. First, from $X$ = "Tom$_5$", $Y$ = "Ken$_6$", or "taiketsu$_2$", we choose $Y$ = "Ken$_6$" that appears last in the document as the root node. Next, from "Ken$_6$" to the beginning of the document, "Tom$_5$" is appended with case marker "others" as a child of the root node, "taiketsu$_2$" is appended with case marker "ga" as a child of the root node, and "Chumoku$_1$" is appended with case marker "others" as a child of "Tom$_5$". Note that the other phrases, such as "mamonaku$_3$" or "hajimaru$_4$", are not appended to the tree because they are not noun phrases.

To discover the extraction patterns of SRL structure, we make a minimal tree that consists of given entities and the candidate, where each phrase is represented by the case marker "CM", an entity class, and the string "STR" and POS of the candidate (See Figure 4.3 (b)).

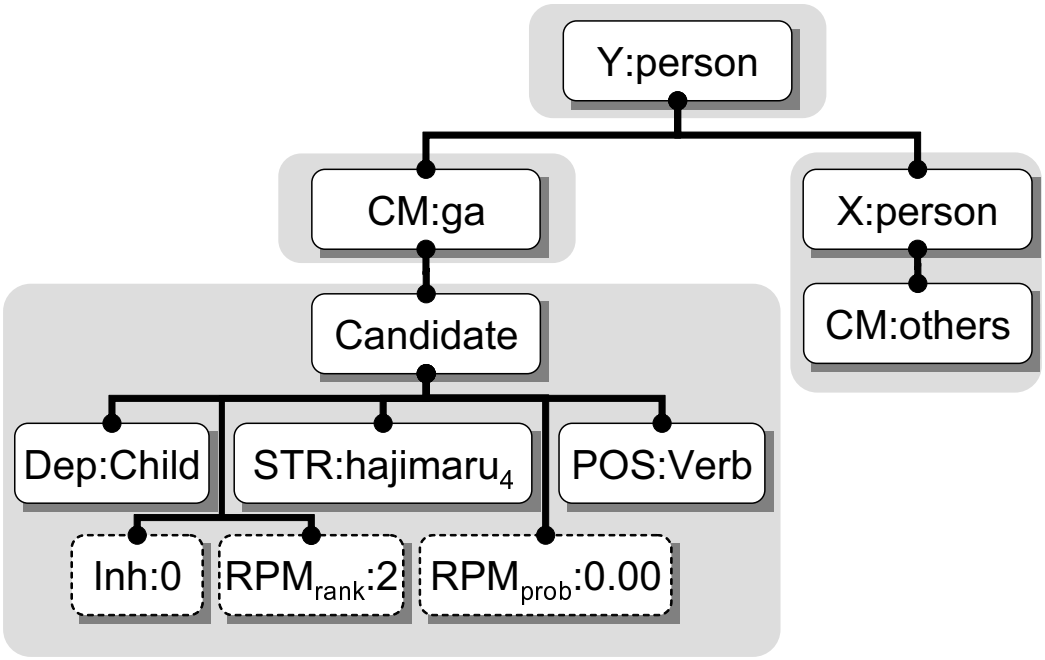


Figure 4.4. The inter-sentential feature tree of $X$="Tom$_5$", $Y$="Ken$_6$", and candidate="hajimaru$_4$" in the document, id = 008.

This approach can only create feature trees of the candidate noun phrases, which are appended to the SRL. For the candidate verb phrases, we make different feature trees from those of the candidate noun phrases using both SRL and dependency structures. We assume that the verb phrases that have referential noun phrases, as parent or child, in a dependency structure must be referential too. Thus, candidate verb phrases are appended as descendents of these noun phrases that are in the SRL using the syntactic representation of "parent" and "child" in the feature tree. For example, given the entity pair, $X$ = "Tom$_5$" and $Y$ = "Ken$_6$", and the candidate, "hajimaru$_4$" in the document, id = 008, we make the feature tree shown in Figure 4.4 using the dependency structure, "hajimaru$_4$" has child node "taiketsu$_2$", which places it in the SRL.

### 4.2.2 Proposed Features

To solve the problems described in Section 4.1, we propose features based on the inherent clues of words and a relation predicting model for recognizing intra-sentential or inter-sentential relation expressions.

#### Inherent Feature

Some words express the relationships between named entities but some do not. For example, the word "mother" suggests a relationship, but the word "car" does not. A list of words that can express relationships between named entities would be useful for recognizing the relation expression of a given pair. As far as we know, however, no such list exists in Japanese. Thus, we estimate which words are able to express relationships between named entities. Here, we assume that all verbs are able to express relationships, and accordingly we focus on nouns.

When relation expression $R$ of entity pair $X$ and $Y$ is a noun, it is possible to say "$Y$ is $R$ of $X$" or "$Y$ is $X$'s $R$". Here, we say that noun $R$ takes argument $X$. In linguistics, this kind of noun is called a relational noun. Grammatically speaking, a relational noun is used to describe relationships just as prepositions do, because its meaning describes a "relation" rather than a "thing". To estimate which nouns are able to express relationships between named entities, we use the distribution characteristic of relational nouns. In linguistics, many researchers have described the relationship between possessives and relational nouns (Barker, 2008). Based on these observations, we use the knowledge found in Tanaka et al. (1999) which states that for patterns "$B$ of $A$" or "$A$'s $B$", if word $B$ is a relational noun, the corresponding word A belongs to a certain semantic category. In contrast, if word $B$ is not a relational noun, the word $A$ belongs to many semantic categories. Figure 4.5 shows the distribution of the semantic categories of "mother" and "car" acquired in the following way.

First, we acquired A and B using the patterns "$A$ no $B$"[1] from a large

[1] "$B$ of $A$" or "$A$'s $B$" in English.

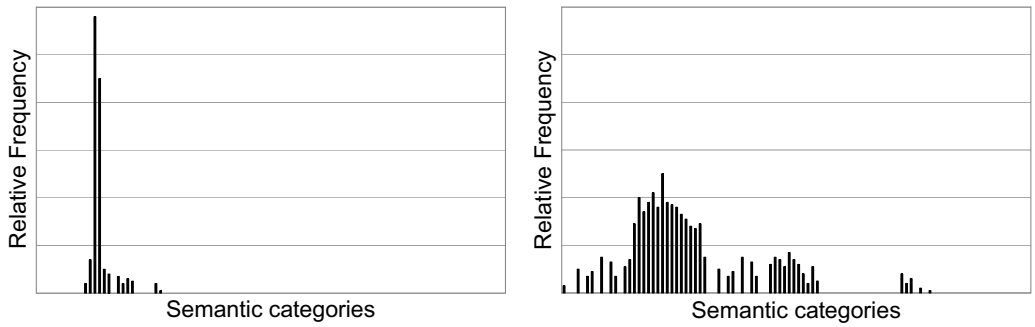


Figure 4.5. Distribution of semantic category of "mother" (left) and "car" (right).

Japanese corpus, and then mapped each $A$ into semantic categories $C$= $\{ c_1, c_2, \cdots, c_m \}$ using a Japanese lexicon (Ikehara et al., 1999). Next, for each $B$, we calculated agglomeration score $\mathcal{H}c(B)$ using the semantic category of corresponding words $A$. When corresponding words $A$ are mapped to $i$ semantic categories, $i > 1$, we counted all mapped semantic categories as $1/i$. Finally, we estimated whether a word is a relational noun by using $k$-NN estimation with positive and negative examples. The result, inherent feature, "Inh:1" shows that it is a relational noun while "Inh:0" shows that it is not. In both cases, the result is appended to the feature tree as a child of the candidate node (See Figure 4.2 (b), 4.3 (b), or 4.4).

$$\mathcal{H}c(B) = -\sum_{c \in C} P(c|B) \log_m P(c|B)$$

where

$$P(c|B) = \frac{freq(c, B)}{freq(B)}$$

In our experiments, we acquired 55,412,811 pairs of $A$ and $B$ from 1,698,798 newspaper articles and 10,499,468 weblogs, which were obtained by a web crawler from July 1, 2004 to Jun 30, 2006. As training data, we used the relation expressions in a training corpus as positive examples and the rest words as negative examples. Then, 446,010

Table 4.1. The evaluation result of the estimation over 400 nouns.

| Accuracy [%] | Precision [%] | Recall [%] |
|---|---|---|
| 81.0 ( 324/400 ) | 71.0 ( 82/116 ) | 66.0 ( 82/124 ) |

nouns were estimated as to whether they were relational nouns or not. Table 4.1 shows the evaluation result of the estimation over 400 nouns that were randomly selected. Accuracy of the estimation is 81.0%, precision is 71.0% and recall is 66.0%. For example, the following relational nouns were correctly estimated, "husband", "wife", "mother", "brother", "president", "employee", "fellow", "partner", and "rival".

**Relation Predicting Model**

There are a number of relationships that typically change as time passes, such as "dating" ⇒ "engagement" ⇒ 'marriage" between people. For instance, if the past relationships of a given pair are "dating" and "engagement" and one candidate is "marriage", "marriage" would be predicted as the relation expression of the given pair. Therefore, the past relations of the given entity pair and the typical relationship changes that occur over time would be useful in recognizing the relation expression of a given pair.

In this thesis, we represent the typical relationship changes that occur over time by the simple relational trigger model, $P_T(r_n|r_m)$. Note that $r_m$ is a past relation and $r_n$ is a relation that changes over time from $r_m$. To make the trigger model, we automatically extracted relation information [$X$, $Y$, $R$, $D$] from newspaper articles and weblog texts that had time stamps of document creation. Using this relation information with time stamps for each entity pair, we sorted the relations in order of time and counted the pairs of present and previous relations. Here, pairs whose present and previous relations occur on the same date were not counted. For example, if we extract "dating" for an entity pair on January 10, 1998, "engagement" on February 15, 2001, and "marriage" on December 24, 2001, the pairs $\langle r_m$ = dating, $r_n$

Table 4.2. Examples of calculated relational trigger model between entity classes.

| Entity Class | $r_m$ | $r_n$ | $P_T(r_n\|r_m)$ | $Count(r_m, r_n)$ |
|---|---|---|---|---|
| person | | dating | 0.050 | 102 |
| ↕ | dating | marriage | 0.050 | 101 |
| person | | engagement | 0.040 | 82 |
| person | | marriage | 0.157 | 786 |
| ↕ | engagement | engagement | 0.065 | 325 |
| person | | wedding | 0.055 | 276 |
| person | | president | 0.337 | 17,081 |
| ↕ | vice president | vice president | 0.316 | 16,056 |
| organization | | CEO | 0.095 | 4,798 |
| organization | | alliance | 0.058 | 8,358 |
| ↕ | alliance | accommodated | 0.027 | 3,958 |
| organization | | acquisition | 0.027 | 3,863 |
| location | | mutual consultation | 0.022 | 2,670 |
| ↕ | neighbour | support | 0.015 | 1,792 |
| location | | visit | 0.012 | 1,492 |
| location | | war | 0.077 | 78,170 |
| ↕ | war | mutual consultation | 0.015 | 15,337 |
| location | | support | 0.010 | 10,226 |

= engagement⟩, ⟨$r_m$ = dating, $r_n$ = marriage⟩, and ⟨$r_m$ = engagement, $r_n$ = marriage⟩ are counted. The counted score is then summed in terms of entity class pairs and the trigger model is calculated by the following formula.

$$P_T(r_n|r_m) = \frac{Count(r_m, r_n)}{\sum_{r_n} Count(r_m, r_n)}$$

For the experiments, we extracted relation information by named entity recognition (Suzuki et al., 2006), relation detection (Hirano et al., 2007), and the proposed method based on the inherent feature described before in this section. A total of 10,463,232 relation information were extracted from 8,320,042 newspaper articles and weblog

texts with time stamps made between January 1, 1991 and June 30, 2006 sourced from Mainichi newspaper and crawled newspaper articles and weblog texts. As examples of the calculated relational trigger model, Table 4.2 shows the top three relations, $r_n$, of several relations, $r_m$, for Japanese standard named entity classes defined in the IREX workshop[2]. For instance, given the pair *person* and *organization*, the relationship "vice president" is most likely to be replaced, over time, by the relationship "president". For our aim, to use the knowledge obtained for recognizing relation expressions in a document, we consider that the relational trigger model would be useful.

To obtain the past relationships of a given pair in the input document, we again used the relation information with time stamps extracted as above. The only relations we used as past relations, $R_m = \{r_{m_1}, r_{m_2}, \cdots, r_{m_k}\}$, are those of a given pair whose time stamps are older than the input document. Finally, the following formula selects the maximum probability from those calculated from past relationships $R_m$ and the trigger model $P_T(r_n|r_m)$.

$$P_T(r_n|R_m) \simeq \max\{P_T(r_n|r_{m1}), P_T(r_n|r_{m2}), \cdots, P_T(r_n|r_{mk})\}$$

Using this calculated probability, we ranked candidates and appended the rank "RPM$_{rank}$" and the probability score "RPM$_{prob}$" to the feature tree as a child of the candidate node (See Figure 4.2 (b), 4.3 (b), or 4.4). For example, if the past relationships were "dating" and "engagement" and candidates are "marriage", "engagement", "meeting', or "eating", the candidates were ranked in terms of probability as "marriage" (RPM$_{prob}$ : $0.15$, RPM$_{rank}$ : $1$), "engagement" (RPM$_{prob}$ : $0.06$, RPM$_{rank}$ : $2$), etc.

To investigate the performance of the relation predicting model, we calculated test set perplexity per relation expression of the model. We randomly divided the extracted 10,463,232 relations in half to create training and test sets. We used the training set to calculate the re-

[2]http://nlp.cs.nyu.edu/irex/

lational trigger model[3]. Finally, we calculated test set perplexity, $PP$, with the following formula. Note that $R_m$ is a set of past relations of $r_n$.

$$PP = 2^H$$

where

$$H = -\frac{1}{N}\sum_{n=1}^{N} \log_2 P_T(r_n|R_m)$$

The perplexity score of the relation predicting model was 14.483 over 1,961,374 relation expressions that have past relations in the test set. For comparison, we also calculated the perplexity of a unigram model, $P_U(r)$, which is the probability of occurrence in the training set.

$$P_U(r) = \frac{freq(r)}{\sum_r freq(r)}$$

The perplexity score of the unigram model was 856.316 over 1,961,374 relation expressions, which are the same relation expressions as those above. These results support our assumption that there are a number of relationships that typically change as time passes.

### 4.2.3 Classification Algorithms

The fundamental idea is to use structural information of texts to discover domain independent extraction patterns. Thus, we use structure-based learning algorithms that have good reported performance from among several learning algorithms that use structural information, such as Tree kernel (Collins and Duffy, 2002), HDAG kernel (Suzuki et al., 2003), or boosting-based algorithm (Kudo and Matsumoto, 2004). The experiments tested Kudo and Matsumoto's boosting-based algorithm using sub-trees as features; it has comparatively short learning times and has been implemented as a BACT system.

[3]In calculating perplexity, to solve the zero frequency problem, we did linear smoothing with an incoming unigram model, $P_U(r)$, whose weight was 0.9 vs. 0.1.

Given a set of training examples, each of which is represented as a tree labeling whether the candidate is the relation expression of a given pair or not, the BACT system learns a set of rules that is effective in classification. Then, given a test instance, the BACT system classifies it using the set of learned rules.

In the classification, to judge which named entity is the subject, $X$, of candidate relation expressions, we classify a test instance by a model that treats the first named entity of the pair as the subject, and a model treats the second named entity as the subject.

## 4.3. Experiments

We conducted experiments using texts from Japanese newspaper articles and weblog texts to test the proposed method against both intra- and inter-sentential tasks. In the experiments, we compared the following methods:

**Conventional Feature:** trained by features based on dependency structure for intra-sentential, and discourse structure for inter-sentential tasks.

**Inherent Feature:** trained by Conventional Feature plus the feature based on inherent clues of candidate words.

**Past Relations Feature:** trained by Inherent Feature plus features based on past relations with a cache model (Kuhn and Mori, 1990). We evaluated this method to provide a reference for Relation Predicting Feature, i.e. to show the effectiveness of using relationships that change over time. The cache model is a way to use past relationships without considering relational change in which probability of occurrence in past relations, $P_C(r)$, calculated by the following formula. Note that $CountPR(r)$ is a number of occurrence in the past relations.

$$P_C(r) = \frac{CountPR(r)}{\sum_r CountPR(r)}$$

The rank, based on the candidate probability, was appended to every candidate feature tree. In the example mentioned in 4.2.2, if the past relationships were "dating" and "engagement" and candidates are "marriage", "engagement", "meeting', or "eating", the candidates were ranked in terms of probability as "engagement" (PR$_{prob}$ : $0.50$, PR$_{rank}$ : $1$), "marriage" (PR$_{prob}$ : $0.00$, PR$_{rank}$ : $2$), etc.

**Relation Predicting Feature:** trained by Inherent Feature plus the feature based on a relation predicting model.

### 4.3.1 Settings

We took 6,200 documents from Japanese newspapers and weblogs dated from January 1, 2004 to June 30, 2006, which were obtained by a web crawler, and manually annotated the relation expressions between named entities for experimental purposes. The named entity pairs targeted in this data were person ↔ person, person ↔ organization, person ↔ location, organization ↔ organization, organization ↔ location, and location ↔ location. In the texts, a total of 17,228 semantically related pairs had already been annotated, so we gave annotators the instruction, "Select any phrase that does not cross a bunsetsu boundary that contains words that express a relationship between the given pair. If a relation expression belongs to more than one bunsetsu, select the phrase that belongs to the last bunsetsu of the relation expression. After selecting a phrase, judge which named entity is the subject, $X$, of the selected phrase." In the data, there were total of 715,655 candidate phrases for 17,228 entity pairs. This means that annotators were selecting phrases from 41.54 phrases on average for each pair.

To investigate inter-annotator agreement, two people annotated 15,005 pairs, a subset of 17,228 pairs. Table 4.3 shows the result. The agree-

Table 4.3. The inter-annotator agreement of 15,005 named entity pairs, *Match*.

| | *Match* [%] |
|---|---|
| person ↔ person | 84.9 ( 3,455 / 4,069 ) |
| person ↔ organization | 89.1 ( 2,543 / 2,853 ) |
| person ↔ location | 88.7 ( 1,814 / 2,044 ) |
| organization ↔ organization | 83.7 ( 1,290 / 1,541 ) |
| organization ↔ location | 88.2 ( 1,229 / 1,393 ) |
| location ↔ location | 86.8 ( 2,696 / 3,105 ) |
| Total | 86.8 ( 13,027 / 15,005 ) |

Table 4.4. The inter-annotator agreement of 15,005 named entity pairs, *Precision* and *Recall*.

| | *Precision* [%] | *Recall* [%] |
|---|---|---|
| person ↔ person | 81.7 ( 2,344 / 2,825 ) | 82.5 ( 2,344 / 2,841 ) |
| person ↔ organization | 86.5 ( 1,409 / 1,628 ) | 84.4 ( 1,409 / 1,669 ) |
| person ↔ location | 87.5 ( 949 / 1,085 ) | 82.2 ( 949 / 1,155 ) |
| organization ↔ organization | 85.8 ( 980 / 1,142 ) | 80.9 ( 980 / 1,211 ) |
| organization ↔ location | 84.8 ( 540 / 637 ) | 79.8 ( 540 / 677 ) |
| location ↔ location | 83.0 ( 1,523 / 1,834 ) | 84.0 ( 1,523 / 1,814 ) |
| Total | 84.2 ( 7,745 / 9,194 ) | 82.7 ( 7,745 / 9,367 ) |

ment of the two annotators was about 86.8% (13,027/15,005). We also calculated precision and recall by assuming that the annotated data produced by person $A$ was the answer, the data produced by person $B$ the system; precision was 84.2% and recall was 82.7% (Table 4.4). The inter-annotator agreement is high as seen above, so we use the annotated data produced by a single person in the experiments.

Table 4.5 shows details of the annotated data produced by a single person for a total of 17,228 entity pairs. relation expressions of 9,178 pairs appeared in the same sentence with the pair, while those of 2,058 pairs appeared in different sentences. The remaining pairs

Table 4.5. Details of the annotated data

| | | # | % |
|---|---|---|---|
| Explicit | Intra-sentential | 9,178 | 53.3 |
| | Inter-sentential | 2,058 | 11.9 |
| Implicit | | 5,992 | 34.8 |
| Total | | 17,228 | |

had no explicit relation expressions. This means that 11,236 phrases were selected out of 715,655 candidate phrases. We used the selected phrases as positive examples and the rest phrases as negative examples. In the intra-sentential experiment, 17,228 entity pairs were given, but only 9,178 of them had relation expressions. In contrast, in the inter-sentential experiment, 8,050 entity pairs (not intra-sentential) were given, and only 2,058 of them had relation expressions.

We conducted five-fold cross-validation over 17,228 entity pairs so that sets of pairs from a single text were not divided into the training and test sets. In the experiments, all features were automatically acquired using a Japanese POS tagger (Fuchi and Takagi, 1998) and dependency parser (Imamura et al., 2007).

### 4.3.2 Results and Discussion

Tables 4.6 and 4.7 show the performance of each method for intra-sentential and inter-sentential extraction. $Precision$ is defined as the percentage of correct relation expressions out of all those extracted. $Recall$ is the percentage of correct relation expressions out of the manual annotation results. The $F$ measure is the harmonic mean of precision and recall. To examine the statistical significance of the results, we used McNemar's paired test, a variant of the sign test, to assess the extraction disagreement. The tables also include the results of significance tests.

Table 4.6. Results of intra-sentential, $Precision$, $Recall$, $F$.

| Feature | $Precision$ [%] | $Recall$ [%] | $F$ |
|---|---|---|---|
| Conventional | 63.5 ( 3,436 / 5,411 ) | 37.4 ( 3,436 / 9,178 ) | 47.1 |
| +Inherent | 67.2 ( 4,036 / 6,001 ) | 43.9 ( 4,036 / 9,178 ) | 53.1 |
| ++Past Relations | 67.5 ( 4,042 / 5,987 ) | 44.0 ( 4,042 / 9,178 ) | 53.3 |
| ++Relation Predicting | 70.7 ( 4,462 / 6,314 ) | 48.6 ( 4,462 / 9,178 ) | 57.6* |

We used McNemar's paired test to assess extraction disagreement. Underlined results indicate that there is a significant difference ($p < 0.01$) against the Conventional Feature. If results are significantly better ($p < 0.01$) against the Inherent Feature, the results are marked by an asterisk.

Table 4.7. Results of inter-sentential, $Precision$, $Recall$, $F$.

| Feature | $Precision$ [%] | $Recall$ [%] | $F$ |
|---|---|---|---|
| Conventional | 70.1 ( 579 / 825 ) | 28.1 ( 579 / 2,058 ) | 40.1 |
| +Inherent | 77.1 ( 719 / 932 ) | 34.9 ( 719 / 2,058 ) | 48.0 |
| ++Past Relations | 74.3 ( 732 / 985 ) | 35.5 ( 732 / 2,058 ) | 48.1 |
| ++Relation Predicting | 75.3 ( 795 / 1,056 ) | 38.6 ( 795 / 2,058 ) | 51.1* |

We used McNemar's paired test to assess extraction disagreement. Underlined results indicate that there is a significant difference ($p < 0.01$) against the Conventional Feature. If results are significantly better ($p < 0.01$) against the Inherent Feature, the results are marked by an asterisk.

**Effects of Inherent Feature**

Comparing the Conventional Features to Inherent Features for intra-/inter-sentential tasks indicates that the proposed method based on inherent clues of words improved intra-sentential task performance, $F$, by 6.0 points and inter-sentential task performance, $F$, by 7.9 points. The significance tests showed the effectiveness of the proposed method (in both tasks, $p < 0.01$). The proposed method correctly recognized the relation expressions of 1,664 pairs that the Conventional Feature could not. For example, "naiyashu$_2$" was estimated as a relational noun and was correctly recognized as the relation expression between the pair, $X$ = "Taigasu$_1$" and $Y$ = "Tanaka$_3$", in the document, id = 009, by the

proposed method; the Conventional one erroneously recognized "happyo$_5$-shita".

**document id = 009**

Taigasu$_1$-wa naiyashu$_2$-no Tanaka$_3$-ga intai$_4$-suru-to happyo$_5$-shita.
(The Tigers$_1$ announced$_5$ that infielder$_2$ Tanaka$_3$ would retire$_4$.)

This result supports our idea that it is useful to accumulate the words that inherently express relationships for recognizing relation expressions between named entities.

**Effects of Relation Predicting Model**

Comparing Inherent Feature to Relation Predicting Feature showed that the proposed method, which is based on a relation predicting model, improved intra-/inter-sentential task performance by 4.5 and 3.1 points, respectively. Significance tests also showed the effectiveness of the proposed method (in both tasks, $p < 0.01$). The proposed method correctly recognized relation expressions of 1,075 pairs that the Inherent Feature could not. For example, "kantoku$_4$" was correctly recognized as the relation expression for the pair, $X$ = "Taigasu$_1$" and $Y$ = "Okada$_3$", in the document, id = 010, by the proposed method; the Inherent one erroneously recognized "ninmei$_6$-shita". In this example, the past relationships of the pair were $R_m$ = kochi (a coach) so the probability ranking was $P_T$("kantoku$_4$"$|R_m$) = 0.75, $P_T$("ninmei$_6$-shita"$|R_m$) = 0.00.

**document id = 010**

Taigasu$_1$-wa Mayumi$_2$-o Okada$_3$ kantoku$_4$-no konin$_5$-ni ninmei$_6$-shita.
(Tigers$_1$ appointed$_6$ Mayumi$_2$ to be the successor$_5$ to Manager$_4$ Okada$_3$.)

To show the effects of using typical relationships that change over time, we also used Past Relations Feature for comparison. Tables 4.6 and 4.7 show that the Relation Predicting Feature performed better

than the Past Relations one. The significance tests showed that there was a significant difference between the methods. The reason for the superior performance of Relation Predicting Feature is that the proposed method correctly recognized the relation expressions that did not appear in the past relations of a given pair.

Thus, we can conclude that using the past relations between a given pair and typical relationships that change over time will help to recognize relation expressions between named entities. We also found that there are no significant differences in accuracy between Inherent Feature and Past Relations features ($p > 0.2$).

### 4.3.3 Error Analysis

The above experiments showed that our proposed features, which are based on the inherent clues of candidate words and a relation predicting model, are effective in recognizing relation expressions between named entities. However, the experiments identified some challenging problems that need to be overcome to improve the method further. Here, we elucidate the remaining problems by analyzing the two main types of errors.

#### Recognized Errors

In 1,852 instances, the method output the wrong relation expressions. Most errors occurred when the named entities were widely separated in the dependency structure. For example, in the document, id = 011, "tsuma$_3$" is the correct relation expression between the pair, $X$ = "Suzuki$_1$" and $Y$ = "Masako$_4$". However, the method recognized "hakken$_6$" even the system knew the relational noun "tsuma$_3$".

**document id = 011**

Suzuki$_1$-san-no ie$_2$-de tsuma$_3$-no Masako$_4$-san-ga noto$_5$-o hakken$_6$.
(At Suzuki$_1$'s house$_2$, (his) wife$_3$, Masako$_4$, found$_6$ a note$_5$.)

To solve this problem, the method needs to take into account of clues to find the arguments of relational nouns (e.g."tsuma$_3$"), which are different from the clues that allow us to find the arguments of verbs.

**Recognition Failure**

In 3,598 instances, the method output no relation expressions. The main cause of this failure was that the sentence holding the named entities had no verb. For example, in the document, id = 012, "enjiru$_5$" is the correct relation expression between the pair, $X$ = "Satoko$_7$" and $Y$ = "Jane$_8$".

**documet id = 012**

Atarashii$_1$ dorama$_2$-dewa Hide$_3$-ga Tom$_4$-o enjiru$_5$.
Soshite$_6$ Satoko$_7$-ga Jane$_8$-o.
(In a new$_1$ drama$_2$, Hide$_3$ performs$_5$ Tom$_4$. And$_6$ Satoko$_7$ (does) Jane$_8$.)

In this example, the relation expression was not present in sentence holding named entities. One solution is to identify the missing verb; a task that remains rather difficult. We also found that some pairs co-occur in a document and tend to share the same relationship. Therefore, recognizing relation expressions of given pairs from a single document at the same time would improve the performance of the method.

## 4.4. Related Work

Recently, open information extraction, a novel domain-independent extraction paradigm, has been suggested (Shinyama and Sekine, 2006; Banko and Etzioni, 2008).

Several clustering-based methods have been proposed to extract semantically-related named entity pairs, $X$ and $Y$, and their relation expressions, $R$, from a large corpus in structured form [$X$, $Y$, $R$] without predefined relations (Hasegawa et al., 2004; Shinyama and Sekine,

2006; Bollegala et al., 2010). These methods extract multiple words around named entities in a document as features and cluster named entity pairs based on distributional similarity of the features. The methods output named entity pairs belonging to the same clusters that have the same relations. They also output key features of the cluster as relation expressions. These clustering-based methods extract relations between named entities from a corpus rather than a document.

Several methods have been introduced to extract relation expressions, $R$, between a named entity pair, $X$ and $Y$, from a given document, $D$, in structured form [$X$, $Y$, $R$, $D$] without predefined relations (Banko and Etzioni, 2008; Zhu et al., 2009). The idea of these methods is to discover domain independent extraction patterns.

Banko proposed a self-supervised learning method using conditional random fields to extract a relation expression from words located between a given pair (Banko and Etzioni, 2008). Zhu proposed a bootstrapping method using Markov logic networks to extract a relation expression from words located between a given pair (Zhu et al., 2009). The point of these methods is that they use word-based features to discover domain independent extraction patterns.

Our method also extracts relation expressions between a given named entity pair from a given text and solves the problem that structural information of texts is rather infrequent in a number of samples by using not only the structural information of texts but also inherent clues of candidate words and the relation predicting model.

## 4.5. Conclusion

The relation expression recognition task is recognizing a relation expression that demonstrates the explicit relation between the semantically related pair co-occur in the document. In a preliminary observation, we found that word-based features, even syntactic ones, were rather infrequent in a number of samples. To solve the above problem, we propose a supervised learning method using two kinds of external

information about candidates acquired from large text corpora automatically. One is lexical information with selected nouns that indicate relations. The other is relation predicting model which predicts present relations between named entities on the basis of past relations of the pair. Our experiments demonstrated that the method improves the $F$ measure by 10.5 points in the intra-sentential task and 11.0 points in the inter-sentential task, and thus helps to recognize relation expressions between named entities.

CHAPTER

# 5

# RELATION ESTIMATION

## 5.1. Introduction

The relation estimation task is estimating the relationship that exists between the semantically related pair that has an implicit relation. The task also includes judging which named entity is the subject of the relation. For example, suppose we would like to estimate a relationship between a semantically related pair, [Minshuto$_1$, Ichiro Yamada$_2$], in the following document.

**document id = 002**

Minshuto$_1$-no Yamada Ichiro$_2$-wa Yamada Jiro$_3$-no ani$_4$-desu.
( Ichiro Yamada$_2$, the Democratic Party$_1$, is Jiro Yamada$_3$'s brother$_4$. )

However there are no relation expressions in the document between the pair, [Minshuto$_1$, Ichiro Yamada$_2$], the implicit relation "member" should be estimated. Then, "Ichiro Yamada$_2$" should be judged as

the subject of the relation "member". As a result, relation information [Ichiro Yamada$_2$, Minshuto$_1$, member, 002] is extracted from the document, id = 002. Note that the implicit relation can be read from a given document. So, even people know the pair has a relationship "president", the relationship should not be estimated when the relationship can not be read from the document. In other words, implicit relations can be read from the document without real world knowledge.

In previous work, several supervised approaches using similarity of noun phrases were proposed to estimate implicit relations between noun phrases (Shimazu et al., 1986; Kurohashi and Sakai, 1999; Srikumar et al., 2008). The idea of these methods is that similar noun phrase pairs have the same relations. There are many related work (Lin and Pantel, 2001; Hasegawa et al., 2004; Bollegala et al., 2010; Ritter et al., 2010) based on the idea and they proposed several similarity measures of named entity pairs, such as DIRT and LinkLDA, on the basis of extracted huge relation information.

However, these similarity measures were rather infrequent in case of ambiguous named entities because they actually use only a named entity in the pair to disambiguate the other named entity. For example, suppose we would like to estimate the implicit relation between the pair, [Nihon$_2$, Kankoku$_3$], in the following document.

**document id = 012**

> World Baseball Classic$_1$-no Nihon$_2$Kankoku$_3$-ga tanoshimi$_4$-desu.
> (I) am excited$_4$ about Japan$_1$ (versus) Republic of Korea$_2$ in World Baseball Classic$_1$.

Nihon$_2$ and Kankoku$_3$ are ambiguous named entities, which are basically assigned to locations by following the definition of the IREX workshop[1].

However, in the above document, these named entities are sport teams. With similarity measures of named entity pairs on the basis of relation information, it is difficult to assign these named entities to

[1]http://nlp.cs.nyu.edu/irex/

sport teams. Then relation estimation system outputs the incorrect implicit relation such as Located which is the most frequent relation type between location and location.

To solve the problem, we propose the similarity measure combining similarity of named entity pairs and similarity of document in which the pair appeared. In the above example, the topic of the document is easily found as SPORT using entire words in the document, such as "Baseball". In our proposed method, using this combined similarity measure, we estimate the implicit relation with $k$-nearest neighbor. First, the method is finding the nearest $k$ examples from labeled data to a given pair. Then, the system outputs the most frequent relation in selected examples as implicit relation between a given pair.

## 5.2. Related work

Several supervised approaches using similarity of noun phrases have been proposed to estimate implicit relations between noun phrases to date (Shimazu et al., 1986; Kurohashi and Sakai, 1999; Srikumar et al., 2008). As implicit relation types, Shimazu et al. (1986) defined about 80 relation types, such as possession, whole-part, purpose or instruments. While, Kurohashi and Sakai (1999) defined only five relation types, such as obligate cases or possession. Even though definitions of implicit relation types vary, the common idea of these methods is that similar noun phrase pairs have the same relations. For instance, Kurohashi and Sakai (1999) first identified the class of noun phrases using a thesaurus and then estimated implicit relation types between pairs of identified classes.

To calculate similarity of named entities, several similarity measures of named entities have been proposed (Lin and Pantel, 2001; Hasegawa et al., 2004; Bollegala et al., 2010; Ritter et al., 2010), such as DIRT and LinkLDA, on the basis of extracted explicit relations for the purpose of paraphrasing or selectional preference. As the similarity measure of named entities, the state-of-the-art similarity measure was

proposed by Ritter et al. (2010) with LinkLDA framework.

Ritter et al. (2010) presents a series of topic models for the task of computing selectional preferences. These models vary in the amount of independence they assume between $X$ and $Y$. At one extreme is IndependentLDA, a model which assumes that both $X$ and $Y$ are generated completely independently. On the other hand, JointLDA, the model at the other extreme assumes both arguments of a specific extraction are generated based on a single hidden variable $Z$. LinkLDA lies between these two extremes and LinkLDA was reported as the best model for extracted relation information. Comparing JointLDA and LinkLDA, instead of imposing a hard constraint that $z_1 = z_2$ in JointLDA, LinkLDA simply assigns a higher probability to states in which $z_1 = z_2$, because both hidden variables are drawn from the same (sparse) distribution $\theta_r$.

## 5.3. Estimating Implicit Relations between Named Entities

To estimate implicit relations between named entities, our proposed method employs the framework of $k$-nearest neighbor to use the idea of the previous work that similar noun phrase pairs have the same relations. The method, first, finds the nearest $k$ examples from labeled data to a given pair using similarity measure. In order to judge which named entity is the subject, $X$, of relations, we calculate two kinds of similarities: one similarity is the first named entity of the pair as the subject, and the other is the second named entity as the subject. Then, the system outputs the most frequent relation in selected examples as implicit relation between a given pair.

For example, given a pair, [Nihon$_2$, Kankoku$_3$] in the document, id = 012, the method finds the nearest $k$ examples from labeled data to the given pair using the novel similarity measure. Suppose that following examples are selected when $k$ = 3: [Japan, Canada, opposition, XXX]

(Nihon$_2$ is subject), [Tigers, Giants, opposition, YYY] (Nihon$_2$ is subject) and [Tokyo, Japan, located, ZZZ] (Kankoku$_3$ is subject). Then, the most frequent relation, "opposition", in the examples as implicit relation between the pair, and "Nihon$_2$" is judged as the subject of the relation. As a result, the system outputs the relation information [Nihon$_2$, Kankoku$_3$, opposition, 012].

The point of the method is how to calculate the similarity measure between a given pair and labeled examples. In this thesis, we propose a novel similarity measure combining similarity of named entity pairs and similarity of document in which the pair appeared.

In this section, we describe the proposed similarity measure, a combination of two similarity measures based on Latent Dirichlet Allocation (LDA). As the similarity measure of named entities, we use the state-of-the-art similarity measure proposed by Ritter et al. (2010) as the basis of relation information with LinkLDA. In addition to, as the similarity measure of documents, we use a similarity measure proposed by Blei et al. (2003) with LDA. We start by describing the similarity measure of documents with LDA.

### 5.3.1 Similarity of documents

To calculate the similarity of documents, our proposed method employs LDA model (Blei et al., 2003), a kind of topic model, which has reported good performance on collaborative filtering or Document classification.

The LDA model is represented as a probabilistic graphical model in Figure 5.1. There are three levels to the LDA representation. The parameters $\alpha$ and $\beta$ are corpus-level parameters, assumed to be sampled once in the process of generating a corpus. The variables $\theta_d$ are document-level variables, sampled once per document. Finally, the variables $Z_{dn}$ and $W_{dn}$ are word-level variables and are sampled once for each word in each document.

It is important to distinguish LDA from a simple Dirichlet multinomial clustering model. A classical clustering model would involve a two-level model in which a Dirichlet is sampled once for a corpus, a

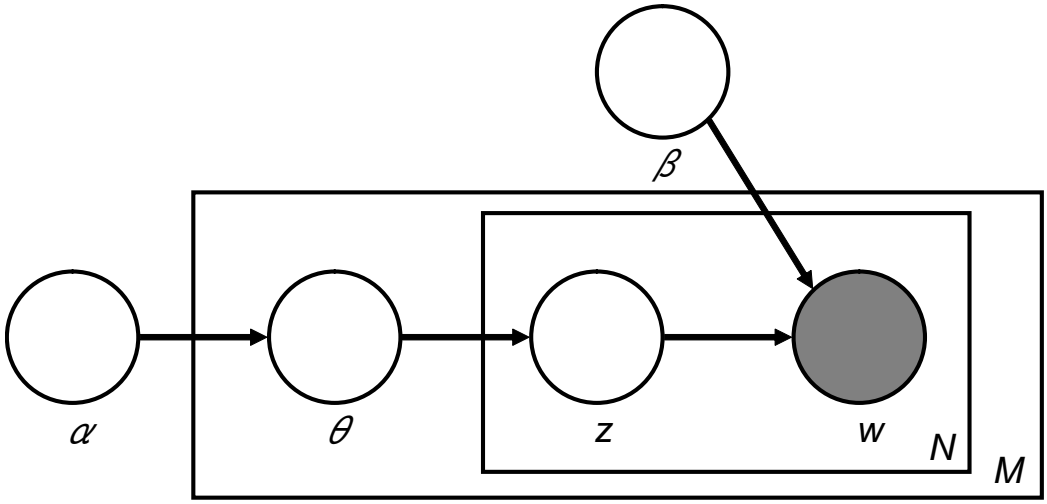


Figure 5.1. Graphical model representation of LDA. The boxes are "plates" representing replicates. The outer plate represents documents, while the inner plate represents the repeated choice of topics and words within a document.

multinomial clustering variable is selected once for each document in the corpus, and a set of words are selected for the document conditioned on the cluster variable. As with many clustering models, such a model restricts a document to be associated with a single topic. LDA, on the other hand, involves three levels, and notably the topic node is sampled repeatedly within the document. Under this model, documents can be associated with multiple topics.

As a similarity measure of documents, we use Jensen Shannon Divergence between the topic distributions of documents. Jensen Shannon Divergence is an information-theoretic measure of the similarity between two probability distributions, and defined as follows.

$$JS(P||Q) = \frac{1}{2}KL(P||M) + \frac{1}{2}KL(Q||M)$$

where

$$KL(P||Q) = \sum_x P(x) log \frac{P(x)}{Q(x)}$$

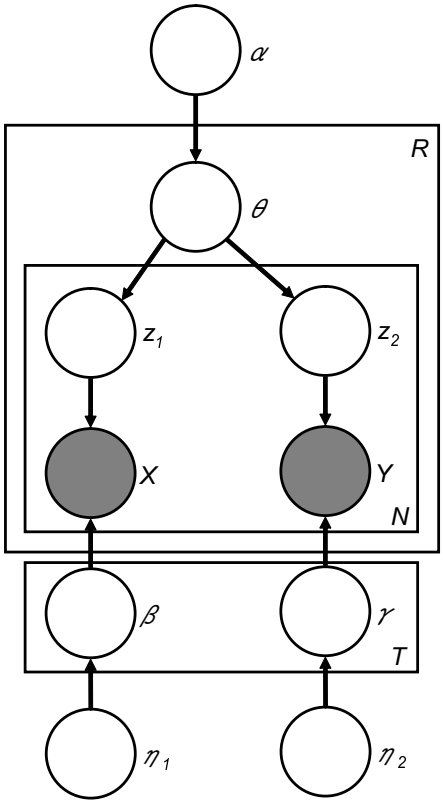


Figure 5.2. Graphical model representation of LinkLDA.

$$M = \frac{1}{2}(P + Q)$$

### 5.3.2 Similarity of Named entities

To calculate the similarity of named entities, our proposed method employs LinkLDA model proposed by Ritter et al. (2010), the state-of-the-art topic model using relation information [$X$, $Y$, $R$, $D$].

Figure 5.2 illustrates the LinkLDA model in the plate notation. The key difference in LinkLDA (versus LDA) is that instead of one, it maintains two sets of topics (latent distributions over words) denoted by $\beta$ and $\gamma$, one for classes of each argument. A topic id $k$ represents a pair of topics, $\beta_k$ and $\gamma_k$, that co-occur in the arguments of extracted re-

lations. The hidden variable $z$ = $k$ indicates that the noun phrase for the first argument was drawn from the multinomial $\beta_k$, and that the second argument was drawn from $\gamma_k$. The per-relation distribution $\theta_r$ is a multinomial over the topic ids.

In particular, note that each $\alpha_i$ is drawn from a different hidden topic $z_i$, however the $z_i$'s are drawn from the same distribution $\theta_r$ for a given relation $r$. To facilitate learning related topic pairs between arguments they employed a sparse prior over the per-relation topic distributions. Because a few topics are likely to be assigned most of the probability mass for a given relation it is more likely that the same topic number $k$ will be drawn for both arguments.

LinkLDA models the generation of both arguments in an extracted relation information. This allows one argument to help disambiguate the other in the case of ambiguous named entities. Also, LinkLDA allows the arguments of a given extraction to be generated from $|Z|$ possible pairs. Thus, LinkLDA simply assigns a higher probability to states in which $z_1$ = $z_2$, because both hidden variables are drawn from the same (sparse) distribution $\theta_r$. LinkLDA can re-use argument classes, choosing different combinations of topics for the arguments if it fits the data better.

As a similarity measure of named entities, we use Jensen Shannon Divergence between the topic distributions of named entities.

### 5.3.3 Combining two Similarities

We combine the two similarity measures of named entities and documents to solve the problem described in Section 5.1. The method employs linear combination with the following formula.

$$Similarity_{combination} = \alpha * Similarity_{namedentity} + (1 - \alpha) * Similarity_{document}$$

The parameter $\alpha$ takes the value between 0 and 1. When $\alpha$ = 1, the method uses only the similarity of named entity. While, when $\alpha$ = 0, the method uses only the similarity of documents.

## 5.4. Experiments

We conducted experiments using texts from Japanese newspaper articles and weblog texts to test the proposed method against implicit relation estimation task. In the experiments, we compared the following methods:

**Baseline:** There is a simple idea that the same pairs of named entity class have the same implicit relations. So, baseline method uses classes of named entity, such as person, organization and location. In the experiments, we use manually annotated named entity classes defined in the IREX workshop. Given a named entity pair, the method selects the labeled examples that have the same pair of classes as the target pair. Then, the system outputs the most frequent relation in selected examples as implicit relation between a given pair.

**Similarity of Documents:** The framework of $k$-nearest neighbor, $k$ = 3, with the similarity of documents described in Section 5.3.1. To calculate the similarity, we give 6,200 documents used in evaluations and 8,320,042 documents sourced from Mainichi newspaper and crawled newspaper articles and weblog texts between January 1, 1991 and June 30, 2006, to a LDA tool[2] with $|Z|$ = 100. As a similarity measure of documents, we use distributional similarity over topics of documents.

**Similarity of Named Entities:** The framework of $k$-nearest neighbor, $k$ = 3, with the similarity of named entities described in Section 5.3.2. To calculate the similarity, we give 12,209,359 relation triples extracted from 8,320,042 documents to a LinkLDA tool[3] with $|Z|$ = 100. These extracted relations are used by named entity recognition (Suzuki et al., 2006), relation detection described in Chapter 3, and the relation expression recognition described

[2]http://chasen.org/~daiti-m/dist/lda/
[3]https://github.com/aritter/LDA-SP/

Table 5.1. Details of the evaluation data

| | | # | % |
|---|---|---|---|
| Explicit | Intra-sentential | 9,178 | 53.3 |
| | Inter-sentential | 2,058 | 11.9 |
| Implicit | | 5,992 | 34.8 |
| Total | | 17,228 | |

in Chapter 4, from documents sourced from Mainichi newspaper and crawled newspaper articles and weblog texts between January 1, 1991 and June 30, 2006. As a similarity measure of named entities, we use distributuional similarity over topics of named entities.

**Combination of two Similarities:** The framework of $k$-nearest neighbor, $k$ = 3, with the combined similarity of documents and named entities described in Section 5.3.3. In the experiments, we change parameter $\alpha$ from 0 to 1 in increments of 0.1.

### 5.4.1 Settings

We took 6,200 documents from Japanese newspapers and weblogs dated from January 1, 2004 to June 30, 2006, which were obtained by a web crawler, and manually annotated the relation expressions between named entities for experimental purposes. The named entity pairs targeted in this data were person ↔ person, person ↔ organization, person ↔ location, organization ↔ organization, organization ↔ location, and location ↔ location. In the texts, a total of 17,228 semantically related pairs had already been annotated. Additionally explicit relation expressions are also annotated for 11,236 pairs (See Table 5.1).

For 5,992 pairs that have implicit relations, we gave annotators the instruction, “Read the document, and write down a relation type of a given pair. Even you know a relationship of the pair, e.g. “president” between “Obama” and “U.S.”, you should not write “president” unless

Table 5.2. The detail of annotated implicit relation types.

| Relation Types | # | Argument $X$ | Argument $Y$ |
|---|---|---|---|
| mates | 2,934 | person | person |
| located | 1,552 | person, organization, location | location |
| member | 1,065 | person, organization, location | organization, location |
| residence* | 236 | location | person |
| play a role* | 76 | person | person |
| opposition* | 70 | person, organization | person, organization |
| publisher* | 59 | organization | person |
| TOTAL | 5,992 | | |

the relationship can be read from the document. After writing a relation type, mark up which named entity is the subject, $X$, of the written relation."

In the annotations, we did not give a list of relation types such as ones on ACE program, to investigate the nature of implicit relations between named entities. In the annotated data produced by a single person for a total of 5,992 named entity pairs, only seven relation types are annotated, such as "mates", "located", "member", "residence", "play a role", "opposition" and "publisher". Table 5.2 shows the details of annotated implicit relation types. However, relation types defined on ACE program covered the top three relation types, relation types marked with asterisk are the types not defined on ACE program. This indicates that relation types defined on previous work is not enough to cover the nature of implicit relation types.

To investigate inter-annotator agreement, two people annotated 1,000 pairs, randomly selected from 5,992 pairs. Table 5.3 shows the result. The agreement of the two annotators was 89.9% (899/1,000). The inter-annotator agreement is high as seen above, so we use the annotated data produced by a single person in the experiments.

We conducted leave-one-out cross-validation over 5,992 entity pairs. In the experiments, all documents were used a Japanese POS tag-

Table 5.3. The inter-annotator agreement of 1,000 relation pairs.

| | *Match* [%] |
|---|---|
| person ↔ person | 92.5 ( 468 / 506 ) |
| person ↔ organization | 95.2 ( 79 / 83 ) |
| person ↔ location | 83.0 ( 146 / 176 ) |
| organization ↔ organization | 76.5 ( 13 / 17 ) |
| organization ↔ location | 84.5 ( 71 / 84 ) |
| location ↔ location | 91.0 ( 122 / 134 ) |
| Total | 89.9 ( 899 / 1000 ) |

Table 5.4. Results of implicit relations, *Accuracy*,

| Methods | *Accuracy* [%] |
|---|---|
| Baseline | 83.2 ( 4,987 / 5,992 ) |
| Similarity of Documents | 77.2 ( 4,623 / 5,992 ) |
| Similarity of Named Entities | <u>91.5</u> ( 5,484 / 5,992 ) |
| Combination of two Similarities ($\alpha$ = 0.7) | <u>96.5</u>$*$ ( 5,785 / 5,992 ) |

We used McNemar's paired test to assess estimation disagreement. Underlined results indicate that there is a significantly better ($p < 0.01$) against the Baseline. If results are significantly better ($p < 0.01$) against the Similarity of Named Entities, the results are marked by an asterisk.

ger (Fuchi and Takagi, 1998) and dependency parser (Imamura et al., 2007).

### 5.4.2 Results and Discussion

Table 5.4 shows the performance of each method for implicit relation estimation. *Accuracy* is defined as the percentage of correct relation types out of all pairs. To examine the statistical significance of the results, we used McNemar's paired test, a variant of the sign test, to assess the estimation disagreement. The table also includes the results of significance tests.

**Effects of Combination of two Similarities**

Comparing the Similarity of Named Entities, the state-of-the-art method, to our proposed method, Combination of two Similarities, for implicit relation estimation task indicates that the proposed method improved performance, $Accuracy$, by 5.0 points. The significance tests showed the proposed method's effectiveness ($p < 0.01$). The proposed method correctly estimated the relation types of 317 pairs that the Similarity of Named Entities could not. For example, relation type "opposition" was correctly estimated between the pair, $X$ = "Taigasu$_1$" and $Y$ = "Yakuruto$_3$", in the document, id = 013, by the proposed method; the Similarity of Named Entities erroneously estimated "member".

**document id = 013**

> Kino$_1$-no Taigasu$_2$･Yakuruto$_3$-wa omoshiro$_4$-katta. Koshien$_5$ saiko$_6$.
> ((The game of) Tigers$_2$ (versus) Yakult$_3$ was excited$_4$ yesterday$_1$. Koshien$_5$ is terrific!)

This result supports our idea that it is useful to combine the similarity of documents into the similarity of named entities for estimating relation types between named entities.

Figure 5.3 shows the perfomance of proposed method changing the parameter $\alpha$ from 0 to 1 in increments of 0.1. When $\alpha$ = 1, the method uses only the similarity of named entity. While, when $\alpha$ = 0, the method uses only the similarity of documents. The figure demonstrates that the proposed method with $\alpha$ = 0.7 is the best $accuracy$, 96.5%.

### 5.4.3 Error Analysis

The above experiments showed that our proposed method, which combines two similarities of named entities and documents, is effective in estimating relation types between named entities. However, the experiments identified some problems that need to be overcome to improve the method further. Here, we elucidate the remaining problems by analyzing the main type of errors.

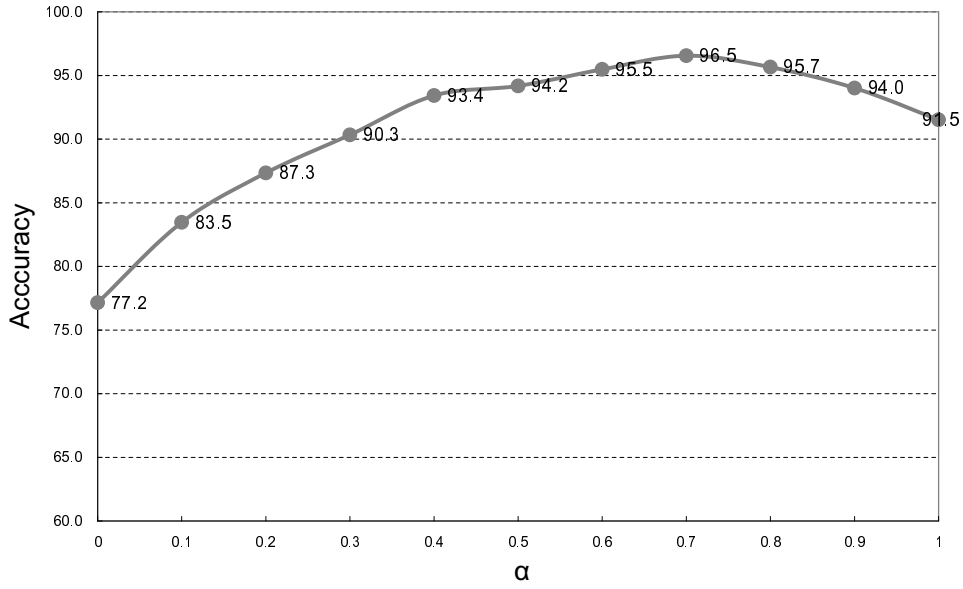


Figure 5.3. The performance of the proposed method.

In 207 instances, the method output wrong relation types. Most errors occurred when the manually annotated relation type is "residence". For example, in the document, id = 014, "residence" is the correct relation types between the pair, $X$ = "Yokosukashi$_1$" and $Y$ = "Yamada$_2$". However, the method estimated "located".

**document id = 014**

Yokosukashi$_1$-no Yamada$_2$-san-ga kekkon$_3$-shita.
(Mr. Yamada$_2$, (lives in) Yokosuka City$_1$, was married$_3$.)

The reason why the system output wrong relation type is that almost all implicit relations between person and location are "located" in annotated data. To solve the problem, using syntactic pattern between the pair would help. For example, the patterns, "$X$ no $Y$" or "$X$ ( $Y$ )", demonstrate special implicit relation, residence, between person and location.

## 5.5. Conclusion

The relation estimation task is estimating the relationship that exists between the semantically related pair that has an implicit relation. In a preliminary observation, we found that similarity measures of named entity pairs, on the basis of extracted huge relation information, were rather infrequent in case of ambiguous named entities because they actually use only a named entity in the pair to disambiguate the other named entity.

To solve the problem, we propose the similarity measure combining similarity of named entity pairs and similarity of document in which the pair appeared. In our proposed method, using this combined similarity measure, we estimate the implicit relation with $k$-nearest neighbor. First, the method finds the nearest $k$ examples from labeled data to a given pair. Then, the system outputs the most frequent relation in selected examples as implicit relation between a given pair. Our experiments demonstrated that the method improves the $Accuracy$ by 5.0 points, and thus helps to estimate relation types between named entities.

CHAPTER

# 6

# CONCLUSION

## 6.1. Summary

In this thesis, to extract semantically-related named entity pairs, $X$ and $Y$, and their relations, $R$, from Japanese documents $D$ in structured form [$X$, $Y$, $R$, $D$], we decomposed the relation extraction task into three tasks. The first is detecting semantically-related pairs from named entity pairs that co-occur in a given document (**relation detection**). The second is recognizing a relation expression that demonstrates the explicit relation between the detected pair from the document (**relation expression recognition**), and the third one is estimating the relationship that exists between a detected pair that has an implicit relation (**relation estimation**).

In the case of document "Ichiro Yamada, the Democratic Party, is Jiro Yamada's brother.", in relation detection task, the pairs [Ichiro Yamada, Jiro Yamada] and [Ichiro Yamada, the Democratic Party] should

be detected as semantically-related ones and [the Democratic Party, Jiro Yamada] should not. Then, in relation expression recognition task, "brother" should be recognized as relation expression for [Ichiro Yamada, Jiro Yamada] and no expression should be recognized for [Ichiro Yamada, the Democratic Party]. At last, in relation estimation task, the relationship "member" should be estimated for [Ichiro Yamada, the Democratic Party]. In the tasks of relation expression recognition and relation estimation, we have to decide not only the relation between the pair but also which named entity in the pair is subject, $X$, of the relation.

The contributions of this thesis are follows. In relation detection task, various supervised learning approaches have been explored (Zelenko et al., 2003; Kambhatla, 2004; Culotta and Sorensen, 2004). They use two kinds of features: syntactic ones and word-based ones, for example, the path of the given pair in the parse tree and the word n-gram between named entities (Kambhatla, 2004). They target only intra-sentential relation detection in which named entity pairs are located in the same sentence, in spite of the fact that about 43.6% of named entity pairs with semantic relations are inter-sentential in Japanese documents. Our proposed method is a supervised learning method using contextual features for detecting a semantic relation between a given pair of named entities, which may be located in different sentences.

In relation expression recognition task, to recognize a relation expression from words located between a given pair in English, the prior work proposed methods using conditional random fields or Markov logic networks using only word-based features (Banko and Etzioni, 2008; Zhu et al., 2009). In a preliminary observation, we found that word-based features, even syntactic ones, were rather infrequent in a number of samples. To solve the above problem, we propose a supervised learning method using two kinds of external information about candidates acquired from large text corpora automatically. One is lexical information with selected nouns that indicate relations. The other

is relation predicting model which predicts present relations between named entities on the basis of past relations of the pair. We show that the proposed method outperformed the prior method through the experiments in Japanese corpus.

In relation estimation task, previous work used the idea that similar noun pairs must have the same relations to estimate implicit relation between not only named entities but also general nouns (Shimazu et al., 1986; Kurohashi and Sakai, 1999; Srikumar et al., 2008). To calculate similarity of named entities, several similarity measures of named entities on the basis of extracted huge relation information have been proposed for the purpose of paraphrasing or selective preference (Lin and Pantel, 2001; Hasegawa et al., 2004; Bollegala et al., 2010; Ritter et al., 2010). However, to estimate implicit relations, these similarity measures were rather infrequent in case of ambiguous named entities because they actually use only a named entity in the pair to disambiguate the other named entity. To solve the problem, we propose the similarity measure combining similarity of named entity pairs and similarity of document in which the pair appeared.

## 6.2. Future work

There still remain several topics to explore. We will leave the following questions open to future work.

First, however we proposed the method to extract relation information from documents in this thesis, we have not evaluated the method by the application that using the extracted relation information. We would like to reveal the following question by application based evaluation.

- Is the number of extracted relation information enough?
- Is precision of extracted relation information enough?
- What kind of applications is satisfied with the number and precision of extracted relation information?

Second, to use extracted relation information effectively, synonym resolutions of named entity and relation expression are important (Yates and Etzioni, 2009; Hasegawa et al., 2004; Bollegala et al., 2010; Ritter et al., 2010). For named entity synonym resolution, it is more useful to use not only connection among expressions in documents but also connection between expression and real world entity. External databases such as geometric database and product database can be used. Several methods have been proposed to disambiguate named entities using the above databases, called Grounding. We would like to incorporate these methods into our overall systems.

Third, however we proposed the method to extract relation information from documents in this thesis, the experiments identified some challenging problems that need to be overcome to improve the method further. For example, in relation detection task, the method did not detect semantically-related pairs liked by definite nouns phrases, such as “shusho (the prime minister)” or “shacho (the president)”. In relation expression recognition task, the method output no relation expressions because the sentence holding the named entities had no verb. In relation estimation task, most errors occurred when the manually annotated relation type is “residence”, because that almost all implicit relations between person and location are “located” in annotated data. We need to solve the challenging problems to improve the method further.

# ACKNOWLEDGEMENTS

Many people gave me insightful comments and suggestions. I would like to express my gratitude to them in Japanese.

まず主指導教官の松本裕治教授に心より感謝いたします．松本先生と会わなければ，きっと博士号を取得しようと考えなかったと思います．先生の研究に対する姿勢，学生を育てようとする心，研究者としてとても尊敬すると共に，松本研出身者であることを誇りに感じています．

お忙しい中副指導教官になっていただいきました中村哲教授に深く感謝いたします．中間発表にて中村先生から研究内容を面白いとコメントいただき，とても嬉しかったです．研究をまとめるにあたりとても自信になりました．同じく，副指導教官になっていただいきました新保仁准教授，小町守助教にはとても感謝いたします．特に小町さんには，とてもお世話になりました．お忙しいのに，学外にいる私からのわがままを快諾くださり本当にありがとうございました．また研究に関しても的確なコメントをいただきありがとうございました．

言語処理研究者としての基礎を教えてくださいました，乾健太郎教授，藤田篤さん，飯田龍さんに深く感謝いたします．皆さんに出会えたことを本当に嬉しく思います．また秘書の北川さんにも研究活動や学校生活を暖かく支えていただきました．心から感謝いたします．

この博士論文にまとめた研究は，日本電信電話株式会社にて取り組んできまし

た研究成果です．本研究を進めるにあたり，社内外の皆様から支援いただきました．特に，菊井玄一郎さん，松尾義博さん，浅野久子さん，牧野俊朗さんには，研究の検討時点から論文化まで本当に多くのことを教えていただきました．また小林のぞみさんには，研究で行き詰ったときに相談に乗って頂きました．小林さんに相談に乗ってもらっていなければ，社会人博士も取得できていなかったと思います．本当にありがとうございました．

最後に，愛する真貴子と桃香へ．まこと桃香の支援がなければ，博士論文を書き上げることができなかったと思う．やりたいことといっぱいあったと思うけど，論文執筆を優先してくれて本当にありがとう．とても感謝しています．これからも毎日，まこと桃香の素敵な笑顔をみれるようにがんばります．

# REFERENCES

Agichtein, Eugene and Luis Gravano (2000) 'Snowball: Extracting Relations from Large Plain-Text Collections.' In 'Proceedings of the 5th ACM conference on Digital libraries' pp. 85–94

Banko, Michele and Oren Etzioni (2008) 'The Tradeoffs Between Open and Traditional Relation Extraction.' In 'Proceedings of the 46th Annual Meeting on Association for Computational Linguistics: Human Language Technologies' pp. 28–36

Barker, Chris (2008) *Semantics: An international handbook of natural language meaning* (Walter De Gruyter Inc)

Bhide, Manish A., Ajay Gupta, Rahul Gupta, Prasan Roy, Mukesh K. Mohania and Zenita Ichhaporia (2007) 'LIPTUS: associating structured and unstructured information in a banking environment.' In 'Proceedings of the 2007 ACM SIGMOD international conference on Management of data' pp. 915–924

Blei, David M., Andrew Y. Ng and Michael I. Jordan (2003) 'La-

tent dirichlet allocation.' *The Journal of Machine Learning Research* 3(30), 993–1022

Bollegala, Danushka, Yutaka Matsuo and Mitsuru Ishizuka (2010) 'Relational Duality: Unsupervised Extraction of Semantic Relations between Entities on the Web.' In 'Proceedings of the 19th International World Wide Web Conferece' pp. 151–160

Brin, Sergey (1998) 'Extracting Patterns and Relations from the World Wide Web.' In 'WebDB Workshop at 6th International Conference on Extending Database Technology' pp. 172–183

Bunescu, Razvan, Ruifang Ge, Rohit J. Kate, Edward M. Marcotte, Raymond J. Mooney, Arun Kumar Ramani and Yuk Wah Wong (2005) 'Comparative Experiments on Learning Information Extractors for Proteins and their Interactions.' *Artificial Intelligence in Medicine (special issue on Summarization and Information Extraction from Medical Documents)* (2), 139–155

Cai, Yuhan, Xin Luna Dong, Alon Halevy, Jing Michelle Liu and Jayant Madhavan (2005) 'Personal information management with SEMEX.' In 'Proceedings of the 2005 ACM SIGMOD international conference on Management of data' pp. 921–923

Chakaravarthy, Venkatesan T., Himanshu Gupta, Prasan Roy and Mukesh Mohania (2006) 'Efficiently linking text documents with relevant structured information.' In 'Proceedings of the 32nd international conference on Very large data bases' pp. 667–678

Chakrabarti, Soumen, Jeetendra Mirchandani and Arnab Nandi (2005) 'SPIN: searching personal information networks.' In 'Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval' pp. 674–674

Cohen, William W., Einat Minkov and Anthony Tomasic (2005) 'Learning to understand web site update requests.' In 'Proceedings of

the 19th international joint conference on Artificial intelligence' pp. 1028–1033

Collins, Michael and Nigel Duffy (2002) 'Convolution Kernels for Natural Language.' *Advances in Neural Information Processing Systems* 14, 625–632

Culotta, Aron and Jeffrey Sorensen (2004) 'Dependency Tree Kernels for Relation Extraction.' In 'Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics' pp. 423–429

Cunningham, Hamish, Diana Maynard, Kalina Bontcheva and Valentin Tablan (2002) 'GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications.' In 'Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics'

Cutrell, Edward and Susan T. Dumais (2006) 'Exploring personal information.' *Commun. ACM*

Doorenbos, Robert B., Oren Etzioni and Daniel S. Weld (1997) 'A scalable comparison-shopping agent for the World-Wide Web.' In 'Proceedings of the first international conference on Autonomous agents' pp. 39–48

Fuchi, Takeshi and Shinichiro Takagi (1998) 'Japanese Morphological Analyzer using Word Co-occurrence - JTAG.' In 'Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics,' vol. 1 pp. 409–413

Ghani, Rayid, Katharina Probst, Yan Liu, Marko Krema and Andrew Fano (2006) 'Text mining for product attribute extraction.' *SIGKDD Explorations Newsletter* 8(1), 41–48

Grishman, Ralph (1997) 'Information Extraction: Techniques and Challenges.' In 'International Summer School on Information Extrac-

tion: A Multidisciplinary Approach to an Emerging Information Technology' pp. 10–27

Grishman, Ralph, Silja Huttunen and Roman Yangarber (2002) 'Information extraction for enhanced access to disease outbreak reports.' *Journal of Biomedical Informatics* 35(4), 236–246

Grosz, Barbara J., Aravind K. Joshi and Scott Weinstein (1983) 'Providing a Unified Account of Definite Noun Phrases in Discourse.' In 'Proceedings of the 21st annual meeting on Association for Computational Linguistics' pp. 44–50

Hasegawa, Takaaki, Satoshi Sekine and Ralph Grishman (2004) 'Discovering Relations among Named Entities from Large Corpora.' In 'Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics' pp. 415–422

He, Bin, Mitesh Patel, Zhen Zhang and Kevin Chen-Chuan Chang (2007) 'Accessing the deep web.' *Communications on ACM* 50(5), 94–101

Hirano, Toru, Yoshihiro Matsuo and Genichiro Kikui (2007) 'Detecting Semantic Relations between Named Entities in Text Using Contextual Features.' In 'Proceedings of the 45th Annual Meeting on Association for Computational Linguistics' pp. 157–160

Hobbs, Jerry R., John Bear, David Israel and Mabry Tyson (1993) 'FASTUS: A finite-state processor for information extraction from real-world text.' In 'IJCAI' pp. 1172–1178

Ikehara, Satoru, Masahiro Miyazaki, Satoru Shirai, Akio Yoko, Hiromi Nakaiwa, Kentaro Ogura, Masafumi Oyama and Yoshihiko Hayashi (1999) *Nihongo Goi Taikei (in Japanese)* (Iwanami Shoten)

Imamura, Kenji, Genichiro Kikui and Norihito Yasuda (2007) 'Japanese Dependency Parsing Using Sequential Labeling for Semi-spoken Language.' In 'Proceedings of the 45th Annual Meeting on Association for Computational Linguistics' pp. 225–228

Jansche, Martin and Steven P. Abney (2002) 'Information extraction from voicemail transcripts.' In 'Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10' EMNLP '02 pp. 320–327

Kambhatla, Nanda (2004) 'Combining Lexical, Syntactic, and Semantic Features with Maximum Entropy Models for Extracting Relations.' In 'Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics' pp. 178–181

Kudo, Taku and Yuji Matsumoto (2004) 'A Boosting Algorithm for Classification of Semi-Structured Text.' In 'Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing' pp. 301–308

Kuhn, Roland and Renato De Mori (1990) 'A Cache-Based Natural Language Model for Speech Recognition.' *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12(6), 570–583

Kurohashi, Sadao and Yasuyuki Sakai (1999) 'Semantic Analysis of "A NO B "Noun Phrases using a Machine Readable Dictionary.' *IPSJ SIG Notes* 99(2), 109–116

Lin, Dekang and Patrick Pantel (2001) 'DIRT - Discovery of Inference Rules from Text.' In 'Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining' pp. 323–328

Liu, Bing, Minqing Hu and Junsheng Cheng (2005) 'Opinion observer: analyzing and comparing opinions on the Web.' In 'Proceedings of the 14th international conference on World Wide Web' pp. 342–351

Minkov, Einat, Richard C. Wang and William W. Cohen (2005) 'Extracting personal names from email: applying named entity recognition to informal text.' In 'Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing' pp. 443–450

Nariyama, Shigeko (2002) 'Grammar for Ellipsis Resolution in Japanese.' In 'Proceedings of the 9th International Conference on Theoretical and Methodological Issues in Machine Translation' pp. 135–145

Pantel, Patrick and Marco Pennacchiotti (2006) 'Espresso: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations.' In 'Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics' pp. 113–120

Popescu, Ana-Maria and Oren Etzioni (2005) 'Extracting product features and opinions from reviews.' In 'Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing' pp. 339–346

Popowich, Fred (2005) 'Using text mining and natural language processing for health care claims processing.' *SIGKDD Explorartion Newsletter* 7(1), 59–66

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

Riloff, Ellen (1993) 'Automatically constructing a dictionary for information extraction tasks.' In 'Proceedings of the eleventh national conference on Artificial intelligence' pp. 811–816

Ritter, Alan, Mausam and Oren Etzioni (2010) 'A Latent Dirichlet Allocation method for Selectional Preferences.' In 'Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics' pp. 424–434

Shimazu, Akira, Shozo Naito and Hirosato Nomura (1986) 'Analysis of semantic relations between nouns connected by a Japanese particle "No".' In 'Computational Linguistics,' vol. 15 pp. 247–266

Shinyama, Yusuke and Satoshi Sekine (2006) 'Preemptive Information Extraction using Unrestricted Relation Discovery.' In 'Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology' pp. 304–311

Srikumar, Vivek, Roi Reichart, Mark Sammons, Ari Rappoport and Dan Roth (2008) 'Extraction of Entailed Semantic Relations Through Syntax-based Comma Resolution.' In 'Proceedings of the 46th Annual Meeting on Association for Computational Linguistics: Human Language Technologies' pp. 1030–1038

Suzuki, Jun, Erik McDermott and HIdeki Isozaki (2006) 'Training Conditional Random Fields with Multivariate Evaluation Measures.' In 'Proceedings of the 43th Annual Meeting on Association for Computational Linguistics'

Suzuki, Jun, Tsutomu Hirao, Yutaka Sasaki and Eisaku Maeda (2003) 'Hierarchical Directed Acyclic Graph Kernel: Methods for Structured Natural Language Data.' In 'Proceedings of the 41st Annual Meeting on Association for Computational Linguistics' pp. 32–39

Tanaka, Shosaku, Yoichi Tomiura and Toru Hitaka (1999) 'Classification of Syntactic Categories of Nouns by the Scattering of Semantic Categories (in Japanese).' *Transactions of Information Processing Society of Japan* 40(9), 3387–3396

Turmo, Jordi, Alicia Ageno and Neus Català (2006) 'Adaptive information extraction.' *ACM Computer Services*

Wong, Wilson, Wei Liu and Mohammed Bennamoun (2010) 'Acquiring Semantic Relations using the Web for Constructing Lightweight Ontologies.' In 'Proceedings of the 13th Pacific-Asia Conference on Knowledge Discovery and Data Mining'

Yates, Alexander and Oren Etzioni (2009) 'Unsupervised Methods for

Determining Object and Relation Synonyms on the Web.' *Journal of Artificial Intelligence Research* 34, 255–296

Zelenko, Dmitry, Chinatsu Aone and Anthony Richardella (2003) 'Kernel Methods for Relation Extraction.' *Journal of Machine Learning Research* 3, 1083–1106

Zhu, Guangyu, Timothy J. Bethea and Vikas Krishna (2007) 'Extracting relevant named entities for automated expense reimbursement.' In 'Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining' pp. 1004–1012

Zhu, Jun, Zaiqing Nie, Xiaojing Liu, Bo Zhang and Ji-Rong Wen (2009) 'StatSnowball: a Statistical Approach to Extracting Entity Relationships.' In 'Proceedings of the 18th international conference on World Wide Web' pp. 101–110

# LIST OF PUBLICATIONS

## Journal Papers

1. Toru Hirano, Yoshihiro Matsuo and Genichiro Kikui, Detecting Semantic Relations between Named Entities Using Contextual Features, Journal of Natural Language Processing, Vol. 15, No. 4, pp. 43-58, 2008.09. (in Japanese)

2. Toru Hirano, Ryu Iida, Atsushi Fujita, Kentaro Inui and Yuji Matsumoto, Augmenting a Semantic Verb Lexicon with a Large Scale Collection of Example Sentences, Journal of Natural Language Processing, Vol. 13, No. 3, pp. 113-132, 2006.07. (in Japanese)

## International Conferences and Workshops

1. Toru Hirano, Hisako Asano, Yoshihiro Matsuo and Genichiro Kikui, Recognizing Relation Expression between Named Entities based on Inherent and Context-dependent Features of Relational words,

Proceedings of the 23rd International Conference on Computational Linguistics, pp. 409-417, 2010.

2. Toru Hirano, Yoshihiro Matsuo and Genichiro Kikui, Aggregating Knowledge of Named Entity Relations, NSF Sponsored Symposium on Semantic Knowledge Discovery, Organization and Use, 2008.

3. Toru Hirano, Yoshihiro Matsuo and Genichiro Kikui, Detecting Semantic Relations between Named Entities in Text Using Contextual Features, Proceedings of the 45th Annual Meeting on Association for Computational Linguistics, pp. 157-160, 2007.

4. Kentaro Inui, Toru Hirano, Ryu Iida, Atsushi Fujita and Yuji Matsumoto, Augmenting a Semantic Verb Lexicon with a Large Scale Collection of Example Sentences, Proceedings of the 5th International Conference on Language Resources and Evaluation, pp. 365-368, 2006.

## Other Publications

1. Toru Hirano, Toshiro Makino and Yoshihiro Matsuo, Estimating Purchasings from Consumer Generated Media, Proceedings of the 18th Annual Meeting of the Association for Natural Language Processing, pp. 187-190, 2012. (in Japanese)

2. Toru Hirano, Yoshihiro Matsuo and Genichiro Kikui, Recognizing Relation Expression between Named Entities, Proceedings of the 15th Annual Meeting of the Association for Natural Language Processing, pp. 921-924, 2009. (in Japanese)

3. Genichiro Kikui, Yoshihiro Matsuo, Nozomi Kobayashi, Toru Hirano and Hisako Asano, Rich Annotation: Knowledge Extraction focusing on Named Entities, IEICE technical report. Natural language understanding and models of communication, Vol. 108(141), pp. 73-78, 2008. (in Japanese)

4. Toru Hirano, Yoshihiro Matsuo and Genichiro Kikui, Location Disambiguation using Geographic Distance and Popularity, Proceedings of the 70th National Convention of Information Processing Society of Japan, Vol. 2, pp. 85-86, 2008. (in Japanese)

5. Toru Hirano, Yoshihiro Matsuo and Genichiro Kikui, Detecting Semantic Relations between Named Entities in Text, Proceedings of the 13th Annual Meeting of the Association for Natural Language Processing, pp. 115-118, 2007. (in Japanese)

6. Toru Hirano, Ryu Iida, Atsushi Fujita, Kentaro Inui and Yuji Matsumoto, Large-scale Example-assignment to Argument Structures in Verb Dictionary, Proceedings of the 11th Annual Meeting of the Association for Natural Language Processing, pp. 396-399, 2005. (in Japanese)