Exploiting World Knowledge in Discourse Processing

– A Comparison of Feature-Based and Inference-Based Approaches –

Naoya Inoue

Graduate School of Information Sciences

Tohoku University

A thesis submitted for the degree of

Doctor of Information Science

January 2013

Acknowledgements

I would like to thank Professor Kentaro Inui whose comments and suggestions were of inestimable value for my whole study. He is the chief examiner for my thesis and has made his support available in a number of ways. I also would like to thank Professor Akinori Itoh and Professor Takeshi Tokuyama. They are the sub examiner of my thesis. They gave me a lot of insightful comments on my study and advised me how to improve my thesis.

I am also indebted to Associate Professor Naoaki Okazaki who gave me invaluable comments and warm encouragements. This thesis would not have been possible without Assistant Professor Yotaro Watanabe. He gave me many advices on an abductive learning framework in terms of statistical machine learning perspective.

I am deeply grateful to Dr. Jerry Hobbs and Dr. Ekaterina Ovchinnikova at Information Sciences Institute (ISI) at University of Southern California. When I had been a visiting researcher at ISI for six months in 2011 and 2012, they gave me a lot of useful and insightful comments on my work. My experience at ISI is unforgettable and valuable. I believe that a lot of achievements in this thesis would not have been possible if I couldn't collaborate with them.

I would also like to express my gratitude to my family for their moral support and warm encouragements. I gratefully appreciate the financial support of Japan Society of Promotion Sciences Fellowship that made it possible to complete my thesis.

I am indebted to my colleagues in our lab to support me. Every time I got depressed, they always smiled me and gave me a Red Bull. Especially, I would like to thank Yamaki-san, the secretary of our lab.

I believe that our lab would go bankrupt if she was not the secretary of our lab. Finally, I owe my deepest gratitude to my computers, buildings, people, people around the world, everything around me, and a campus at Graduate School of Information Sciences, Tohoku University in Japan.

Abstract

Discourse processing is the task of making implicit information explicit in natural language texts. Typically, when the writer makes information implicit, they think that the reader shares the same *world knowledge*, or the knowledge about our world including commonsense knowledge, with the writer. For machines, therefore, it is not an easy task to recover the information omitted by the human writer. To bridge the knowledge gap between humans and machines, a machine-readable database of world knowledge (*knowledge base*) is expected to play an important role in discourse processing.

What is a computational mechanism that enables us to effectively exploit the knowledge base for discourse processing? In the last few years, a number of techniques that acquire world knowledge resources have been developed. As a result, a number of machine-readable resources have been made available. Given the recent advances of statistical approaches to NLP, most of the existing approaches to discourse processing encodes the world knowledge as the feature vectors for machine learning-based classifiers (*feature-based approaches*).

However, a feature-based approach has a severe limitation when it is used as a framework of discourse processing. The limitation is that the feature-based approaches cannot derive implicit information from a text by *combining* several kinds of world knowledge. In discourse processing, it is crucial to have the capability of deriving new information through the combination of different types of world knowledge.

Let us consider the case of *coreference resolution*, one of the discourse processing tasks where we identify the group of linguistic expressions that refers to the same real-world entity. Conventional approaches

to coreference resolution have exploited world knowledge to capture syntactic or semantic compatibility between mentions, encoding them as a feature vector for machine learning-based classifiers. However, there exist many cases where several antecedents are syntactically or semantically compatible with an anaphoric expression, and the derivation of implicit information through the combination of several world knowledge provides a key solution to these problems.

In order to address this issue, we explore an *inference*-based approach, based on a mode of inference called *abduction*, or inference to the best explanation. Specifically, we adopt *cost-based abduction* on first-order logic, where the plausibility of explanation is evaluated through a real-valued function (*cost function*). In first-order logic abduction, world knowledge is encoded as a set of logical forms: world knowledge is *declaratively* used in the inference-based approach. The declarative encoding of world knowledge naturally overcomes the limitations of feature-based approaches, and enables us to infer the most plausible, implicitly stated information combining heterogeneous inference rules and the pieces of information observed from texts.

In spite of successful theoretical progress and small-scale systems, work on large-scale, "real life" systems foundered on two main difficulties so far: (i) reasoning procedures were not efficient enough, and (ii) the cost function is hand-tuned for each task. In this thesis, we propose an efficient inference method of cost-based abduction in first-order predicate logic that avoids computationally expensive grounding procedures in order to explore inference-based approaches in realistic settings. Through the large-scale evaluation, we demonstrate that the proposed procedure outperforms the previous approaches.

We then show how to formulate the supervised machine learning problem of abduction with the framework of online large-margin training, which has been shown to have both predictive performance and scalability to larger problems. We demonstrate that the proposed training framework successfully reduces the predictive loss in both open tests

and closed tests.

Using the proposed inference and learning frameworks, we give an in-depth comparison of two approaches in both qualitative and empirical perspectives. Specifically, we conduct a case study on anaphora resolution, where we create two machine learning-based anaphora resolution models following feature-based and abductive inference-based approaches. We propose a machine learning-based hybrid model that combines the conventional compatibility feature-based approach with a logical inference-based approach. We integrate those two approaches to complement the weakness of each approach, using an abductive inference framework. The empirical evaluation and the qualitative analyses demonstrate that inference-based approaches have several potential advantages to feature-based approaches.

Contents

List of Figures

Chapter 1

Introduction

In natural language texts, the writer frequently omits the information that the reader can restore by using commonsense knowledge or taking the plausible interpretation of the given texts into account. For example, in the sentences "*John bought a car. The engine was good.*", the writer does *not* explicitly mention that *the engine* is the engine of the car that John bought, because the writer expects that the reader can identify it with the commonsense knowledge that "engine is a part of car", and the plausible interpretation that the engine is part of John's car.

Discourse processing, a subtask of natural language processing (NLP), is the process of making implicit information, such as the information above, explicit in natural language texts. In this thesis, we study the computational aspect of discourse processing, i.e. what kind of computational mechanism should be built for the realization of automatic discourse processing. The study of computational aspect of discourse processing is essential to a broad range of scientific researches, such as cognitive science and linguistics, because the natural language is the basic communication tool that people use. Moreover, we could exploit automatic discourse processing to extract some useful information from a vast amount of texts online, which are produced by the recent advances of Information Technology represented by World Wide Web.

The omission of information in a text is basically triggered by the writer's assumption that the reader shares the same *world knowledge*, the knowledge about our world including commonsense knowledge, with the writer. However, com-

puters do *not* have world knowledge; thus it is not an easy task for computers to restore the information omitted by the human writer. There could be several ways to bridge the gap of world knowledge between humans and machines. In this thesis, we take the most straightforward solution, which gives a *knowledge base*, a machine-readable database of world knowledge, to computers. In order to exploit the knowledge base in automatic discourse processing, we need to address the following two big issues.

Firstly, how do we construct the knowledge base? The amount of world knowledge seems unlimited and growing everyday, but is it possible to construct the knowledge base large enough to approximate the actual world knowledge? In the last few years, a number of techniques that acquire world knowledge resources have been developed. As a result, a number of machine-readable resources have been made available that encode the kinds of knowledge needed for NLP [Fellbaum, 1998; Ruppenhofer et al., 2010, etc.]. Moreover, statistical corpus-based methods for extracting general rules from large amounts of text have been devised [Chklovski and Pantel, 2004; Hovy et al., 2011; Penas and Hovy, 2010; Schoenmackers et al., 2010, etc.]. Therefore, in this thesis, we assume that the knowledge base is sufficient enough to explore the computational aspect of discourse processing, and do *not* address the issue of how to acquire the knowledge base.

The second issue is what computational mechanism enables us to exploit the knowledge base in an effective way. Given the recent advance of statistical approaches to NLP, most of the existing approaches for NLP encodes the world knowledge as the feature vectors of machine learning-based classifiers, which are designed for each particular NLP task (henceforth, we call it *feature-based* approaches). However, a feature-based approach has a severe limitation when it is used as a framework of discourse processing. The limitation is that the feature-based approaches cannot derive implicit information from a text by *combining* several kinds of world knowledge. In discourse processing, it is crucial to have the capability of deriving new information through the combination of different types of world knowledge.

Let us consider the case of *coreference resolution*, one of the discourse processing tasks where we identify the group of linguistic expressions that refers to the

same real-world entity. Conventional approaches to coreference resolution have exploited world knowledge to capture syntactic or semantic compatibility between mentions, encoding them as a feature vector for machine learning-based classifiers. However, there exist many cases where several antecedents are syntactically or semantically compatible with an anaphoric expression, and the derivation of implicit information through the combination of several world knowledge provides a key solution to these problems. Consider the following example:

- *The scientists gave the chimps some bananas because they were hungry.*

We have two antecedents *[the scientists]*$_{j'}$ and *[the chimps]*$_j$, which are both semantically compatible with the anaphor *[they]*$_j$. Therefore, the compatibility feature-based methods cannot choose the correct antecedent *[the chimps]*$_j$ in a systematic way.

Thus, the question remains as to which mechanism enables world knowledge to be maximally effective in discourse processing. This is the main research issue in this thesis. In the next section, we elaborate on how to address this issue in our study.

1.1 Research Issues and Methodologies

To find out what mechanism is able to effectively exploit world knowledge, we work on the main hypothesis that an *inference*-based approach would be a better alternative mechanism to feature-based approaches. Specifically, we explore a first-order logical inference-based discourse processing framework, based on a mode of inference called *abduction*, or inference to the best explanation. Logic-based abductive discourse processing has been studied intensively in the 1980s and 1990s; Hobbs et al. [1993] show that the lowest-cost abductive explanation provides the solutions to a broad range of natural language understanding problems, such as word sense disambiguation, anaphora, and metonymy resolution.

In abductive inference-based approaches, world knowledge is encoded as a set of first-order logical formulae, and used as the background knowledge of logical inference. That is, world knowledge is *declaratively* used in inference-based approaches, while world knowledge is *procedurally* used in feature-based approaches.

The declarative encoding of world knowledge brings us two benefits, which are the key advantages of using abduction for discourse processing:

- abduction-based approaches naturally resolve interdependencies between NLP tasks, and identifies the most coherent interpretation to all tasks without writing a specific procedure;
- abduction-based approaches jointly resolve the ambiguity of knowledge applicability, selecting the most plausible knowledge according to a given context;
- abduction-based approaches infer the most plausible, implicitly stated information combining heterogeneous inference rules and the pieces of information observed from texts.

From a machine learning perspective, abduction-based processing amounts to the joint inference of a particular NLP task and derivation of implicit information. The point is that, however, we can naturally model the joint inference without complicated steps. What we have to do is to write an appropriate knowledge base.

In spite of successful theoretical progress and small-scale inference-based systems, work on large-scale, "real life" systems foundered on three main difficulties so far: (i) there was no sufficiently large knowledge base of the right sort for language processing, (ii) reasoning procedures were not efficient enough, and (iii) a function that evaluates the plausibility of explanation was manually tuned for each purpose. As mentioned earlier, the difficulty (i) is almost resolved, but the difficulties (ii) and (iii) still remain. Therefore, the inference-based approach has not been able to be evaluated on real-life problems and a large knowledge base so far. This leads us to the following research issue:

1. Do inference-based approaches indeed enable us to exploit world knowledge in a effective way better than feature-based approaches do on a real-life dataset? What is the difference between inference-based approaches and feature-based approaches in both a qualitative and empirical perspective?

To answer this question, we conduct the case study on *anaphora resolution* and give a detailed comparison of two approaches from both the qualitative and empirical perspective. Anaphora resolution is the task of identifying the referents of mentions or objects related to mentions, which is one of the important tasks of natural language processing.

Let us return to the three issues of inference-based approaches so far. In abduction, there exists a several explanations to observation in general. To pick the best one, we need some measure to evaluate the plausibility of explanations. In this thesis, we adopt *cost-based abduction*, a variant of abduction where the best explanation is defined as the explanation that minimizes the cost function of explanation. In first-order logical cost-based abduction (henceforth, we call it *first-order cost-based abduction*), the problem of finding the best explanation is equivalent to a constrained combinatorial optimization problem with respect to the cost function, which is an NP-hard problem [Charniak and Goldman, 1991]. In fact, Ovchinnikova et al. [2011] report that Mini-TACITUS abductive reasoning system [Mulkar et al., 2007] could not complete to search the entire search space of explanations within 30 minutes in most of the RTE problems in their experiments.

In the literature, many researchers have tried to overcome cost-based abduction's inefficiency by a range of methods from approximation to exact inference [Chivers et al., 2007; Ishizuka and Matsuo, 1998; Poole, 1993b; Santos, 1996, etc.]. For example, Santos (1994) formulate cost-based abduction in propositional logic using Integer Linear Programming (ILP), and showed its efficiency. However, to the best of our knowledge, most of the proposed methods are optimized for propositional logic. In order to employ these methods for first-order cost-based abduction, we need to transform knowledge bases and observations to propositional logic (henceforth, we call the transformation *grounding*). The process of grounding generates a quite huge search space and does not scale to larger problems. This leads to the second research issue:

2. To maximally receive the benefits of large-scale knowledge acquisition studies, how do we efficiently search for the best explanation in first-order cost-based abduction?

To address this issue, we propose an Integer Linear Programming-based solution to perform an efficient search in first-order cost-based abduction.

Less attention has been paid to the issue (iii), i.e. how to automatically learn the cost function, which rank candidate explanations in order of their plausibility. To apply abductive inference to a wide range of tasks, this non-trivial issue needs to be addressed, as the criterion of plausibility is highly task-dependent. A notable exception is a series of studies in the context of Statistical Relational Learning [Blythe et al., 2011; Kate and Mooney, 2009; Raghavan and Mooney, 2010; Singla and Domingos, 2011], where they emulate abduction in the probabilistic deductive inference framework, Markov Logic Networks (MLNs) [Richardson and Domingos, 2006], or Bayesian Logic Programs [Kersting and Raedt, 2001]. These approaches can exploit several choices of machine learning methods originally developed for probabilistic models [Huynh and Mooney, 2009; Lowd and Domingos, 2007, etc.].

However, emulating abduction in these approaches has severe overhead. For example, the emulation in MLNs requires a special procedure to convert abduction problems into deduction problems because MLNs are deductive inference framework in nature. This conversion process generates a large number of axioms, and hence hampers the application of MLN-based approaches to larger problems (see Sec. 4.5 for more detail). Since inference is a subroutine of learning procedure, learning is also intractable on large dataset, as reported in Singla and Domingos [2011]. This motivates us to address the third research issue:

3. How do we train a cost function of abduction for predicting the desired explanations?

We propose an online large-margin training framework of first-order cost-based abduction based on Passive Aggressive algorithm [Crammer et al., 2006].

1.2 Contributions

In this thesis, we address three research questions described in the previous section, making the following contributions:

- **In-depth comparison of feature-based and inference-based approaches**: We give a detailed comparison of feature-based approaches and inference-based approaches in both qualitative and empirical ways. To the best of our knowledge, none of the previous studies discuss the difference between feature-based approaches and inference-based approaches, nor compare both approaches through a large-scale evaluation (Chapter 2, 5, 6).

- **Lifted inference technique for cost-based abduction on first-order logic**: We propose an efficient method of cost-based abduction in first-order predicate logic that avoids computationally expensive grounding procedures (Chapter 3). Most of the previous approaches to cost-based abduction have solved cost-based abduction in propositional-level. However, it is computationally expensive to convert first-order logic formulae into propositional logic formulae. Our method does not require grounding and works directly on first-order level. We also provide a mathematical proof of the completeness and soundness of the proposed inference procedure.

- **ILP formulation of first-order cost-based abduction problem**: We formulate the best-explanation search problem as an Integer Linear Programming (ILP) optimization problem (Chapter 3). We can exploit several choices of state-of-the-art combinatorial optimization technology developed in Operations Research. In addition, the resulting framework is highly extensible; e.g., we can easily incorporate linguistically motivated heuristics by simply adding some ILP variables and/or constraints to an optimization problem, keeping the overall framework unchanged. We also show how Cutting Plane Inference, which is an iterative optimization strategy developed in Operations Research, can be applied to make first-order abduction more scalable.

- **Supervised learning framework for cost-based abduction**: We show how to formulate the machine learning problem of abduction with the framework of online large-margin training, which has been shown to have both predictive performance and scalability to larger problems (Chapter 4). We generalize the score function of abduction as a weighted linear model and

then learn the weight vector. The linear model-based formulation enables us to flexibly design the score function. We support *partially observed gold-standard explanations* as training examples, where the weights are learned to rank any explanation that includes the gold-standard explanation as the best explanation.

- **Providing the all-in-one software package for cost-based abduction on the public webpage**: We have implemented the proposed techniques of inference and learning in one software package, which is called *Henry*. The software is publicly available at the author's webpage.[1] This is the first all-in-one-package that accomplishes efficient inference and supervised learning.

1.3 Thesis Overview

The rest of this thesis is structured as follows.

- **Chapter 2: Inference-based Approach for Discourse Processing** We introduce an inference-based approach for discourse processing problems. Our hypothesis is that first-order logic-based abduction would be a better approach for solving the discourse processing problems. We first review a history of abductive discourse processing, and then elaborate on how the discourse processing problems can be cast as the abductive inference problem, following the framework of *Interpretation as Abduction* [Hobbs et al., 1993]. We demonstrate that the abduction-based formalism solves several discourse processing problems in an integrated fashion and is also a promising alternative to exploit world knowledge in an effective way.

- **Chapter 3: ILP-based Lifted Inference for Cost-based Abduction** The problem of finding the least-cost abductive explanation is an NP-hard problem. In this chapter, we propose an efficient inference method for cost-based abduction on first-order logic. We show how to perform cost-based abduction directly on first-order level in a similar way to resolution

[1] `http://github.com/naoya-i/henry-n700/`

[Robinson, 1965] and formulate the problem of least-cost explanation finding as an Integer Linear Programming problem.

- **Chapter 4: Online Large-margin Weight Learning for Cost-based Abduction** We address the issue of how to give a cost function that reasonably evaluates the plausibility of explanations. In this chapter, we propose a supervised approach for learning the cost function. To make the learning algorithm possible to have both scalability and generalization ability, we adopt Passive Aggressive algorithm [Crammer et al., 2006], an online large-margin training algorithm.

- **Chapter 5: Resolving Direct and Indirect Anaphora with Feature-based Approach** To take a deeper look at the difference between feature-based and inference-based approaches, we conduct the case study on anaphora resolution. We first propose a feature-based model of anaphora resolution. We give a detailed error analysis of the feature-based anaphora resolution model, and then discuss the problem of feature-based approaches. We found out that it is difficult for the feature-based approaches to solve problems where there are more than one candidate antecedents which are semantically compatible with an anaphor in the preceding context.

- **Chapter 6: Inference-based Approach to Coreference Resolution** Based on the error analyses of previous chapter, we propose an inference-based model for coreference resolution, the subtask of anaphora resolution. Specifically, we propose a machine learning-based hybrid model that combines the conventional compatibility feature-based approach with a logical inference-based approach. We integrate those two approaches to complement the weakness of each approach, using an abductive inference framework. For inference and learning, we use an efficient inference method proposed in Chapter 3 and a learning framework proposed in Chapter 4. We show that inference-based formalisms can infer implicit information that functions as useful clue for coreference resolution. We conduct a large-scale empirical evaluation and demonstrate that the inference-based approach is promising alternative to feature-based approaches.

- **Chapter 7: Conclusions** We summarize our discussion, and present our future direction.

Chapter 2

Inference-based Approach for Discourse Processing

In this chapter, we give a basic idea of how to formalize discourse processing problems in a logical inference-based framework. We first start with the introduction of first-order logic and first-order logic-based abduction. We then describe *cost-based* abduction, where the best explanation is defined as the lowest cost explanation among possible explanations. We then introduce *Interpretation as Abduction* [Hobbs et al., 1993], which is a pioneering work of abductive natural language processing framework.

2.1 First-Order Logic

First-Order Logic (FOL) is a language for meaning representation. In FOL, the basic unit of meaning is an *atom*. An atom is a form of $P(X_1, X_2, ...X_n)$, which consists of two parts: (i) *predicate* P and (ii) *terms* $X_1, X_2, ...X_n$. A predicate is a symbol that represents relation between objects. A term represents object in the world. Therefore, the atom $love(John, Mary)$ means that $John$ and $Mary$ is in a relationship of *love*. Terms can be *constant*, *variable*, or *function* symbols. A constant symbol exactly specifies one object in the world, and a variable symbol means any objects in the world. A function symbol represent mappings from objects to objects.

An atom can be *negated*. When an atom is negated, the truth value of an atom becomes false. In FOL, a negation is represented by "¬", so a negated atom is written as $\neg P(X_1, X_2, ...X_n)$. Negation can be recursively applied, and if negation is applied to an negated atom, then it becomes a non-negated atom. A non-negated atom or negated atom is called *literal*. A *ground atom, or ground literal* refers to an atom or literal where all the terms are constants.

We sometimes want to represent that there are multiple facts being true or false. In FOL, a *logical connector* can be used to connect multiple literals. A logical connector can be $L_1 \wedge L_2$ (*conjunction*, true iff both L_1 and L_2 are true), $L_1 \vee L_2$ (*disjunction*, true iff L_1 or L_2 are true), $L_1 \Rightarrow L_2$ (*implication*, true iff L_1 is false or L_2 is true), or $L_1 \Leftrightarrow L_2$ (*equivalence*, true iff L_1 and L_2 have the same truth value). For example, $love(John, Mary) \wedge [love(John, Catherine) \vee love(John, Ada)]$ means that John loves Mary, *and* John loves Catherine *or* Ada (could be both). *Formula* is an literal or literals that are connected by the logical connectors. The logical connectors can also connect formulas (e.g. $L_1 \wedge L_2 \Rightarrow L_3$).

Variables can be *universally quantified* ($\forall$) or *existentially quantified* ($\exists$) in a formulae. The quantification is written as $\forall x_1, x_2, ...x_n F$ or $\exists x_1, x_2, ...x_n F$, where $x_1, x_2, ...x_n$ is variables that are quantified. Universal quantification means that the formula is true iff the formula is true for all objects in the world. Existential quantification means that the formula is true iff the formula is true for at least one object in the world. A formula is *satisfiable* iff there exists a truth assignment to each literal in the formula which makes the formulae true. A formula F_1 is said to be *entailed* by another formula F_2 iff F_2 is true in every truth assignments which satisfy F_1. The entailment relation is denoted by $\models$.

2.2 Abduction

Abduction is inference to the best explanation. We use function-free first-order logic with finite domains as the meaning representation of abduction in this thesis. Formally, first-order logical abduction is defined as follows:[1]

[1]The same framework is used in *induction*. While induction finds a set of plausible rules from observations, abduction finds a set of plausible facts.

- **Given:** Background knowledge B, and observations O, where B is a set of first-order logical formulae, and O is a set of literals or equalities.
- **Find:** An *explanation* (or *hypothesis*) H such that $H \cup B \models O, H \cup B \not\models \bot$, where H is a set of literals or equalities. Each element in H is called an *elemental explanation*.

Let us define some terminologies. We define *equality* to be the form $x = y$ (*positive* equality) or $x \neq y$ (*negative* equality), where x and y are either variables or constants. The equality $x = y$ means that referents of x and y are the same (i.e. $\{p(x), p(y), x = y\}$ has the same meaning as $\{p(x)\}$). We say that the literal p is the *logical consequence* of S if $S \models p$; p is (*explicitly*) *hypothesized* w.r.t. H if $p \in H$; p is *implicitly hypothesized* if $H \cup B \models p$ w.r.t. H and B (i.e. p is a logical consequence of H w.r.t. B); p is *explained* if $H \cup B \setminus \{p\} \models p$; otherwise p is *assumed*. We refer to the operation that we unify two or more literals in set S of literals, and apply the unifier to S as *factoring* of S.

In this paper, we assume that all variables occurring in a logical form of background knowledge are *universally* quantified with the widest possible scope, unless it is explicitly stated as existentially quantified. On the one hand, we assume that variables occurring in an explanation and observation are *existentially* quantified implicitly. We assume that the background knowledge has no cyclic dependencies between an explaining literal and an explained literal (e.g. $B = \{P(x) \rightarrow Q(x), Q(x) \rightarrow P(x)\}$ has a cyclic dependency). We call this assumption *knowledge recursion-free assumption*.

Typically, several explanations H explaining O exist. We call each of them a *candidate explanation*, and represent a set of candidate explanations of O given B as $\mathcal{H}_{O,B}$. The goal of abduction is to find the best explanation among candidate explanations by a specific evaluation measure.

2.3 Cost-based Abduction

Typically, several explanations H explaining O exist. We call each of them a *candidate explanation*, and represent a set of candidate explanations of O given B as $\mathcal{H}_{O,B}$. The goal of abduction is to find the best explanation among candidate

explanations by a specific evaluation measure. In this paper, we formulate abduction as the task of finding the minimum-cost explanation $\hat{H}$ among $\mathcal{H}_{O,B}$. Henceforth, we refer to abduction based on the minimum-cost explanation finding as the *Cost-based Abduction (CBA)*. Formally, we find $\hat{H} = \arg\min_{H \in \mathcal{H}_{O,B}} cost(H)$, where *cost* is a function $\mathcal{H}_{O,B} \to \mathbb{R}$, which is called the *cost function*.

Let us describe the task of abduction with a toy example. Given $B = \{p(x,y) \wedge q(x) \to r(x), s(x) \to r(x)\}, O = \{r(z)\}$, we have four candidate explanations: $H_1 = \{r(z)\}$, $H_2 = \{p(z,w), q(z)\}$, $H_3 = \{s(z)\}$, and $H_4 = \{p(z,w), q(z), s(z)\}$. The task of abduction is to select the best explanation among them in terms of the cost. Suppose $cost(H_1) = 5.5$, $cost(H_2) = 12.25$, $cost(H_3) = 10.8$, and $cost(H_4) = 7.13$. The correct prediction is then H_1.

In the literature, several kinds of cost functions have been proposed, including cost-based and probability-based [Charniak and Goldman, 1991; Hobbs et al., 1993; Poole, 1993a; Raghavan and Mooney, 2010; Singla and Domingos, 2011, etc.]. In this paper, we adopt the cost function proposed by Hobbs et al. (1993). The cost function assumes that each elemental explanation $p \in H$ has the cost of hypothesizing p (intuitively, the plausibility of p being an explanation for given observations), and sums up the costs of *assumed* elemental explanations. Henceforth, we write $P(x)^{\$c}$ to denote $P(x)$ having a cost c.

During the construction of H, one can *factor* H to generate a new explanation at any time. When H is factored, the following things happen: (i) the literal that has the smallest cost among a set of unified literals remains in H, and (ii) for the unifier $\{x_i/y_i\}_{i=1}^n$, a set of elemental explanations $\{x_i = y_i^{\$0}\}_{i=1}^n$ is added to H. For example, one can factor $H = \{R(a)^{\$20}, R(b)^{\$10}, Q(a)^{\$20}\}$ with the unifier $\{a/b\}$ to get $H' = \{R(b)^{\$10}, a = b^{\$0}, Q(b)^{\$20}\}$, where the smaller cost \$10 is assigned to $R(b)$. Formally, the cost function is defined as follows:

$$cost(H) = \sum_{h \in A(H)} cost(h), \tag{2.1}$$

where $A(H)$ is a set of *assumed* literals in H.

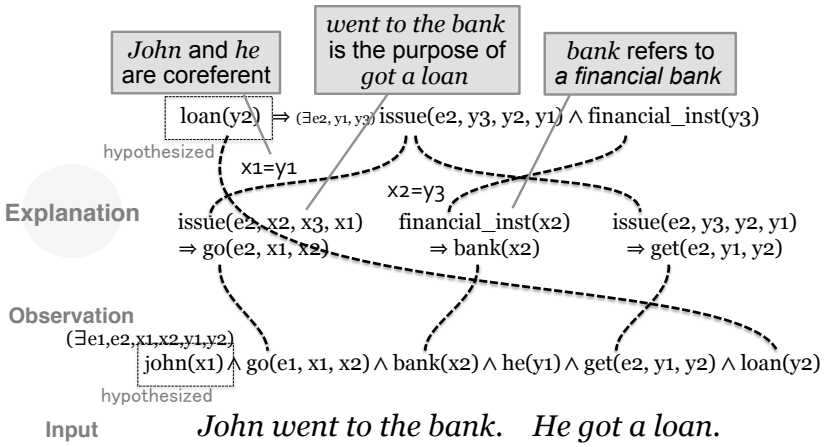


Figure 2.1: Example of abductive interpretation.

2.4 Interpretation as Abduction

Hobbs et al. (1993) pioneered an abduction-based approach for natural language understanding. The key idea is that "*interpreting sentences is to prove the logical forms of sentences, allowing assumptions, merging redundancies where necessary.*" They demonstrate that a wide range of NLP tasks involved in discourse interpretation, including anaphora resolution, discourse relation recognition, etc., can be cast as the problem of finding an explanation to the pieces of information observed from the discourse.

A logical form (LF) of a text represent observations, which need to be explained by background knowledge. In our discourse processing pipeline, a text is first input to the English parser *Boxer* [Bos, 2008]. For each segment, the parse produced by *Boxer* is a first-order fragment of the DRS language used in Discourse Representation Theory [Kamp and Reyle, 1993]. An add-on to *Boxer* converts the DRS into a logical form in the style of Hobbs [1985].

The LF is a conjunction of propositions, which have generalized entity arguments that can be used for showing relationships among the propositions. Hobbs [1985] extends Davidson [1967]'s approach to all predications and claims that corresponding to any predication that can be made in natural language, there is an eventuality. Correspondingly, any predication in the logical notation has an

extra argument, which refers to the "condition", in which that predication is true. Thus, in the logical form $John(e_1, j) \wedge run(e_2, j)$ for the sentence *John runs*, e_2 is a running event by John and e_1 is a condition of j being named "John".

In the context of discourse processing, we call a hypothesis explaining a logical form *an interpretation* of this LF. The interpretation of the text is carried out by an abductive system. The system tries to prove the logical form of the text, allowing assumptions where necessary. Where the system is able to prove parts of the LF, it is anchoring it in what is already known from the overall discourse or from a knowledge base. Where assumptions are necessary, it is gaining new information.

Figure 2.1 shows an example taken from [Hobbs et al., 1993]. In this example, we solve three types of NLP tasks: (i) coreference resolution, e.g. the coreference relation between *John* and *he* ($x1 = y1$), (ii) intent recognition, e.g. the intention of *John* (backward-inference on $go(e1, x1, x2)$ to $loan(y2)$), and (iii) word sense disambiguation, e.g. the meaning of *bank* is not a riverbank, but a financial institution (backward-inference on $bank(x2)$ to $financial\_inst(x2)$, where the inference rule means that "*a financial institution is expressed as bank in a text*).

Specificity of explanations It is crucial to discuss the specificity of explanations. We say that an explanation H is more *specific* than another explanation H' if $H \cup B \models H'$. As discussed in Hobbs et al. [1993], we want to decide the appropriate specificity of an explanation because there are often little evidence (i.e., observation) to support specific explanations. Traditionally, two extreme modes of abduction have been considered. The first is *most-specific abduction*. In most-specific abduction, what we can explain from background knowledge is all explained, which is suitable for diagnostic systems. In diagnostic systems, users might want to know what causes the current situation as much as possible. Some cost-based approaches and probabilistic approaches fall into this group [Charniak and Goldman, 1991; Raghavan and Mooney, 2010, etc.]. The second is *least-specific abduction*. Literally, an explanation is obtained by just assuming observations in this mode. Using only least-specific abduction makes little sense, but as described below, it makes sense if it is combined with most-specific abduction.

In natural language understanding systems, we need both modes at the same time. Adopting only one of these levels is problematic. For example, if we adopt most-specific abduction, the system yields too specific explanation such as "*Bob took a gun because he would rob XYZ bank using a machine gun which he had bought three days ago*." Conversely, if we adopt least-specific abduction, the system assumes just observation, as in "*Bob took a gun because he took a gun*." We thus want to determine the suitable specificity during inference. To the best of our knowledge, Hobbs et al. (1993)'s weighted abduction is only a framework that concerns the appropriateness of explanation specificity. The cost function of weighted abduction naturally handles this by propagating costs of propositions and unification as described in Sec. 3.2.

2.4.1 Weighted Abduction

As mentioned before, abduction needs to select the best hypothesis, and hence this framework also needs to select the best interpretation based on some evaluation measure. Hobbs et al. [1993] propose the cost function that can evaluate two types of plausibility of hypotheses simultaneously: the *correctness* and *informativeness*. The correctness represents how much reliable the contents of information are. The informativeness is how specific the information is. As discussed in Hobbs et al. [1993], the criterion of plausibility is extremely task-dependent. For example, one might want *hating* as the correct explanation of *killing* in story understanding tasks, while *mentally-ill* might be favored in medical diagnostic tasks. One might desire the most specific explanation possible in medical diagnostic systems, whereas one might want less specific explanations in story understanding systems. Therefore, Hobbs et al. [1993] parametrized the cost function in a way that one can construct the cost function that favors more specific and thus more informative explanations, or explanations less specific but reliable in terms of a specific task by altering the parameters. The resulting framework is called *Weighted Abduction*.

In principle, the cost function gives a penalty for assuming specific and unreliable information but rewards for inferring the same information from different observations. To the best of our knowledge, Hobbs et al. (1993)'s weighted ab-

duction is the only framework that concerns the appropriateness of hypothesis specificity. Hobbs et al. (1993) exploit this cost function for text understanding where the key idea is that interpreting sentences is to find the lowest-cost abductive explanation to the logical forms of the sentences in the agreement of correctness-informativeness tradeoff.

2.4.1.1 The basics

In weighted abduction, observations are given with costs, and background axioms are given with weights. It then performs backward-reasoning on each observation, propagates its cost to the assumed literals according to the weights on the applied axioms, and merges redundancies where possible. A cost of interpretation is then the sum of all the costs on elemental hypotheses in the interpretation. Finally, it chooses the lowest cost interpretation as the best interpretation.

Let us first describe the representations used for background knowledge, observations, and hypothesis in weighted abduction:

- **Background knowledge** B: a set of first-order logical formulae whose literals in its antecedent are assigned positive real-valued *weights*. In addition, both antecedent and consequent consist of a conjunction of literals. We use a notation p^w to indicate "a literal p has the weight w." We define $\mathbf{w}_B$ as a *weight vector* of background knowledge B, where i-th component $\mathbf{w}_{Bi}$ corresponds to a weight of a specific literal in a specific axiom (i.e. each component has a one-to-one mapping to each weight in background knowledge).

- **Observations** O: an existentially quantified conjunction of literals. Each literal has a positive real-valued cost. We use a notation $p^{\$c}$ to denote "a literal p has the cost c," and $cost(p)$ to denote "the cost of the literal p."

- **Hypothesis** H: an existentially quantified conjunction of literals. Each literal also has a positive real-valued cost. The cost of H is then defined as $cost(H) = \sum_{h \in H} cost(h)$.

In the Hobbs et al.'s framework, inference procedure is only defined on the formats defined above, although neither formats of B, O nor H are mentioned explicitly.

2.4.1.2 Procedure of weighted abductive inference

Given a weight vector $\mathbf{w}_B$, the cost function of H is defined as the sum of all the costs of elemental hypotheses in H:

$$cost(H; \mathbf{w}_B) = \sum_{h \in P_H} cost(h; \mathbf{w}_B) \tag{2.2}$$

$$= \sum_{h \in P_H} \left[\prod_{i \in chain(h)} \mathbf{w}_{Bi} \right] cost(obs(h)), \tag{2.3}$$

where P_H is a set of elemental hypotheses that are not explained, $chain(h)$ is a set of indices to a literal in axioms that are used for hypothesizing h, and $obs(h)$ is an observed literal that is back-chained on to hypothesize h.

Let us describe how the weighted abduction works. Like logical abduction, H is abductively inferred from O and B, and the costs of elemental hypotheses in H are passed back from O multiplying the weights on the applied axioms in B. When two elemental hypotheses are unified, the smaller cost is assigned to the unified literal. Let us illustrate how these procedure works taking the following axioms and observations as an example:

$$B = \{\forall x (p(x)^{0.3} \wedge q(x)^{0.9} \rightarrow r(x)), \tag{2.4}$$

$$\forall x \exists y (p(y)^{1.3} \rightarrow b(x)), \tag{2.5}$$

$$O = \exists a (r(a)^{\$20} \wedge b(a)^{\$10}) \tag{2.6}$$

A candidate hypothesis that immediately arises is simply assuming O, i.e., $H_1 = \exists a (r(a)^{\$20} \wedge b(a)^{\$10})$, where $cost(H_1) = \$20 + \$10 = \$30$. If we perform backward inference on $r(a)^{\$20}$ using axiom (1), we get $H_2 = \exists a (p(a)^{\$6} \wedge q(a)^{\$18} \wedge b(a)^{\$10})$ and $cost(H_2) = \$34$. As we said, the costs are passed back from $r(a)^{\$20}$ multiplying the weights on axiom (1), and hence $cost(p(a)) = \$20 \cdot 0.3 = \$6$ and $cost(q(a)) = \$20 \cdot 0.9 = \$18$.

If we perform backward inference on both $r(a)$ and $b(a)$ by using axiom (1) and (2), we get another candidate hypothesis $H_3 = \exists a, b (p(a)^{\$6} \wedge q(a)^{\$18} \wedge p(b)^{\$13})$, in which $p(a)^{\$6}$ is unifiable with $p(b)^{\$13}$ assuming that a and b to be identical. In weighted abduction, since the cost of unified literal is given by the smaller

cost, H_3 is refined as $\exists b(q(b)^{\$18} \wedge p(a)^{\$6})$, and $cost(H_3) = \$24$. Considering only these three candidate hypotheses, a solution hypothesis $H^* = H_3$, which has a minimum cost $cost(H_3) = \$24$.

We mentioned that weighted abduction is able to evaluate the specificity of a hypothesis in Sec. 2.4. The mechanism of specificity evaluation is accomplished by the propagation of costs. We can see the working example of this mechanism in the toy problem above: comparing $cost(H_1)$ with $cost(H_2)$ means determining if $r(a)$ should be explained more specifically or not.

2.5 Conclusion

In this chapter, we described the basic idea of abductive inference-based approach for discourse processing. We elaborated on Hobbs et al.'s Interpretation as Abduction framework, the pioneering work of abductive discourse processing. In the Interpretation as Abduction framework, interpreting sentences amounts to proving the logical forms of the sentences. To rank the possible proofs (explanations), Hobbs et al. use the cost function that can evaluate the *correctness* and *informativeness* of explanations, which are both essential to abductive discourse processing.

Chapter 3

ILP-based Lifted Inference for Cost-based Abduction

In order to apply abduction to real-life problems with large-scale knowledge base, we need to address the following issue: *how to search for the best explanation efficiently*. In this chapter, we adopt *first-order logic-based cost-based abduction* (henceforth, first-order cost-based abduction), where we use function-free first-order logic (FOL) as a representation language. In first-order cost-based abduction, the explanation is represented by a set of literals, and the plausibility of explanation is evaluated through the sum of the costs defined on each literal. The best explanation is defined as the lowest-cost explanation. Finding the lowest-cost explanation can be reduced to a constrained combinatorial optimization problem with respect to the cost function, which is an NP-hard problem [Charniak and Goldman, 1991]; this hampers the application of abduction with large knowledge resources to real-life problems. In fact, Ovchinnikova et al. [2011] report that the Mini-TACITUS cost-based abduction system [Mulkar et al., 2007] could not search the entire search space of explanations within 30 minutes in most of the RTE problems in their experiments.

In the literature, many researchers have tried to overcome cost-based abduction's inefficiency by a range of methods from approximation to exact inference [Chivers et al., 2007; Ishizuka and Matsuo, 1998; Poole, 1993b; Santos, 1994, etc.]. For example, Santos (1994) formulate cost-based abduction in propo-

sitional logic using Integer Linear Programming (ILP), and showed its efficiency. However, to the best of our knowledge, most of the proposed methods are optimized for propositional logic. In order to employ these methods for first-order cost-based abduction, we need to transform knowledge bases and observations to propositional logic (henceforth, we call this transformation *grounding*). The process of grounding generates a quite huge search space, and does not scale to larger problems as discussed in Sec. 3.1.

In this chapter, we provide a sound, complete, and scalable solution to first-order cost-based abduction. The key idea is that we solve first-order CBA problems using the *lifted inference* technique, where each inference operation is directly performed on first-order level like resolution [Robinson, 1965]. In principle, this way of problem formulation gives us three benefits:

- we can reduce the search space of candidate explanations in comparison to a grounding approach, because we are able to avoid instantiating FOL formulae with all possible constants;
- the best explanation finding problem can be reduced to the constrained combinatorial optimization problem of first-order literals and/or equalities, meaning that we can exploit several choices of combinatorial optimization technology developed in Operations Research. Specifically, our optimization problem can be naturally formulated as an Integer Linear Programming (ILP) problem, which can be efficiently solved by existing ILP solvers;
- the resulting framework is highly extensible; e.g., we can easily incorporate linguistically motivated heuristics by simply adding some ILP variables and/or constraints to an optimization problem, keeping the overall framework unchanged.

In the rest of this chapter, we first formalize the best explanation finding in first-order CBA using the lifted inference technique, and then describe how to solve it as the ILP optimization problem.

3.1 Lifted First-order Inference for CBA

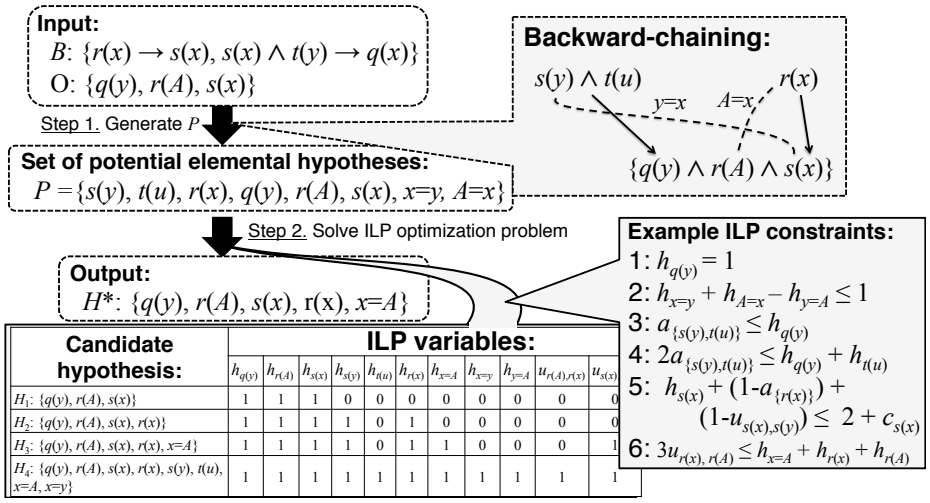


Figure 3.1: Summary of the ILP-based approach.

First-order logic inherits all the theoretical property of propositional logic, and hence the sound and complete inference can be performed on propositional logic. However, performing first-order logical inference on propositional level has severe overhead, because we need *grounding*, which generates the ground instances of first-order logical formulae in knowledge bases and observations (i.e. instantiating them with all possible constants). The grounding procedure generates a large number of formulae when a domain is large. In this chapter, we thus propose to perform the cost-based abduction on first-order level. The approach is in the spirit of resolution [Robinson, 1965], but is applied to the best explanation finding problems. In the rest of this section, we show how to solve the abductive inference problem on first-order level.

Figure 3.1 summarizes our approach. In principle, our approach takes two steps: (i) Step 1: *search-space generation*, and (ii) Step 2: *best-explanation search*. In the search-space generation step, we first construct a set of all possible literals and/or equalities that are potentially included in H. For example, given the toy problem in Sec. 2.3, we construct the following set: $\{r(z), p(z, w), q(z), s(z)\}$. In the best-explanation search step, we find the best explanation for O by finding the best combination of literals or equalities among the set of literals constructed in the search-space generation step, according to the cost function. The problem

is solved in the form of constrained boolean optimization problem, which is the problem of finding the truth assignment to boolean variables that maximizes or minimizes an objective function satisfying the given constraints.

Now we move on to the detail of our approach. To describe the basic idea of our approach clearly, let us restrict the formats of background knowledge, observation, and explanation as follows:

- Background knowledge: a set of first-order Horn clauses (i.e. $p_1 \wedge p_2 \wedge ... \wedge p_n \rightarrow q$, where $p_1, p_2, ..., p_n$, and q are atoms);
- Observations: a set of positive literals or positive equalities;
- Explanation: a set of positive literals or positive equalities.

Henceforth, we call each Horn clause in background knowledge an *axiom*, the right hand side the *head*, and the left hand side the *body*. We show how to extend the expressivity in Sec. 3.3.

We give the overall algorithm in Algorithm 1. Given a background knowledge B and observations O, we first create set P of literals or equalities that are potentially included as constituents of the best explanation of O (line 2–10). We refer to the literal or equality $p \in P$ as the *potential elemental explanation*. To enumerate the potential elemental explanations, we first initialize P with O. We then iteratively apply *backward-inference* to each $p \in P$ (line 2–6). Algorithm 2 depicts the backward-inference operation in detail (line 5–8). We define backward-inference as the following operation:

- **Input:** the Horn clause $p_1 \wedge p_2 \wedge ... \wedge p_n \rightarrow q$ and the literal l, where there must exist the most general unifier θ such that $l\theta = q\theta$.
- **Output:** $\{p_1, p_2, ..., p_n\}\theta$, where the variables that are not substituted by θ are replaced with existentially quantified variables not appearing in P so far.

For example, given the axiom $p(x, y) \wedge q(x, y, z) \rightarrow r(x)$ and $r(a)$, it derives $\{p(a, u_1), q(a, u_1, u_2)\}$, where u_1 and u_2 are existentially quantified variables not appearing in P. Note that P is not equivalent to a set of resolvents that are

Algorithm 1 liftedFirstOrderCBA(Background knowledge B, Observation O, Cost function *cost*)

```
1: P ← O, S ← O
2: while S ≠ φ do
3:    S ← getPotentialElementalExplanations(S, B)
4:    P ← P ∪ S
5: end while
6: for p1, p2 ∈ P do
7:    if ∃θ p1θ = p2θ then
8:       for x/y ∈ θ do P ← P ∪ {x = y}
9:    end if
10: end for
11: return findBestExplanation(P, cost)
```

generated by a particular proof procedure. The goal of proof procedure is to check whether a logical formula is implied by a set of logical formulae. Therefore, the derived proof might not contain a set of *all literals* that can explain observations. For example, SLD resolution [Kowalski, 1974], a backward inference-based proof procedure that works on the Horn clause formulae, what literals resolved upon is selected by a particular computation rule (e.g. leftmost), and the resolution procedure terminates when the proof is found to be failure or success. However, what we want to enumerate is the set of all literals that can explain observations. Since we have the knowledge recursion-free assumption, line 2–11 terminates in a finite time (i.e. until no more backward-inference can be applied).

In line 6–10, we search for the pairs of unifiable literals in P in order to represent the application of factoring operation to H. For each pair of unifiable literals, we add the equalities that are potentially hypothesized by the unifier (see Sec. 2.3). We do *not* unify such literals in P here, because we want to treat that the factoring operations are also defeasible, "*possibly*" true operation. We use the cost function to determine whether they should be factored or not.

In line 11, we find the best explanation. Given P, the problem of best explanation finding can be reduced to a constrained combinatorial optimization problem. Notice that the number of candidate explanations exponentially grows (i.e. $O(2^{|P|})$), because each explanation is represented by the combination of potential elemental explanations. We immediately see that the simple approach

Algorithm 2 getPotentialElementalExplanations(Background knowledge B, set S of literals)

$R \leftarrow \{\}$
for $l \in S$ **do**
  for $p_1 \wedge p_2 \wedge ... \wedge p_n \rightarrow q \in B$ **do**
    if $\exists \theta l\theta = q\theta$ **then**
      for $v \in notSubstitutedVars(\{p_1, p_2, ..., p_n\}, \theta)$ **do**
        $\theta \leftarrow \theta \cup \{v/u_i\}$; $i \leftarrow i + 1$
      end for
      $R \leftarrow R \cup \{p_1, p_2, ..., p_n\}\theta$
    end if
  end for
end for
return R

which finds a minimal explanation by evaluating all the candidate explanations intractable. To improve the inefficiency, we formulate the best explanation finding as the 0-1 ILP optimization problem to exploit the state-of-the-art search strategy of combinatorial optimization problems. The formulation is described in the next section.

3.2 ILP Formulation

We formulate the best-explanation finding problem as an ILP optimization problem, where the search space is represented as ILP variables and constraints, and the cost function is used as the ILP objective. Intuitively, for each $p \in P$, we introduce some 0-1 state variable that represents whether or not the potential elemental explanation p is (explicitly, or implicitly) hypothesized. Then every possible $H \in \mathcal{H}_{\mathfrak{O},\mathfrak{B}}$ can be expressed as the combination of value assignments to these state variables.

We elaborate on two types of ILP variables and ILP constraints in the optimization problem: (i) for representing the search space of candidate explanations, (ii) for implementing the cost function. The ILP formulation here models abduction in FOL without negation. We extend this formulation in Sec. 3.3.1 so that it supports negation.

3.2.1 Formulation for CBA Search Space

To represent whether the literal or equality $p \in P$ is hypothesized (including *implicitly* hypothesized) or not, we introduce an ILP variable $h \in \{0, 1\}$ as follows:

$$\text{for each } p \in P : h_p = \begin{cases} 1 & \text{iff } H \cup B \models p; \\ 0 & \text{otherwise.} \end{cases}$$

For example, H_2 in Figure 3.1 holds $h_{r(x)} = 1$, where $r(x)$ is hypothesized in H_2. We also use h to represent equalities. In H_3, the variable $h_{x=A}$ is set to 1 because $x = A$ is assumed. Note that h variables does not represent the truth values of p (i.e. $h_p = 0$ does not mean $H \cup B \models \neg p$). Once a value assignment to h is determined, we construct H based on the assignment as follows:

Definition 3.2.1 Given a particular value assignment to h variables, we generate an explanation H as follows:

- $p \in H \Leftrightarrow h_p = 1$ for each $p \in P$;
- $p \notin H \Leftrightarrow h_p = 0$ for each $p \in P$.

That is, *all* logical consequences of $H \cup B$ are considered to be an explanation (i.e. the generated explanation includes *implicitly* hypothesized literals, as well as *explicitly* hypothesized literals).

Note that not all value assignments to ILP variables h are allowed. By the definition of candidate explanation in Sec. 2.3, for example, it is not allowed to output the assignment that there exists $p \in O$ s.t. $H \cup B \not\models p$. To ensure that the search space includes only *valid* candidate explanations (i.e. H satisfies $H \cup B \models O$ and $H \cup B \not\models \bot$), we impose several constraints on the value assignments of h. We denote T to represent a set of logical atomic terms in P.

Constraint 1: From the definition of explanation, observations must be the logical consequences of $H \cup B$ (i.e. $H \cup B \models O$).

$$\text{for each } p \in O : h_p = 1 \tag{3.1}$$

Since we assume that P includes only positive literals, it is not required to ensure the consistency of $H \cup B$. In Figure 3.1, the constraint $h_{q(y)} = 1$ is generated.

Constraint 2: From the equality axiom in first-order logic, equality relations must be symmetric (i.e. for all $x, y \in T$, $h_{x=y} = 1 \Rightarrow h_{y=x} = 1$), and transitive (i.e. for all $x, y, z \in T$, $h_{x=y} = 1 \wedge h_{y=z} = 1 \Rightarrow h_{x=z} = 1$). We introduce the following constraints:

$$\text{for each } x, y \in T : h_{x=y} = h_{y=x} \tag{3.2}$$

$$\text{for each } x, y, z \in T : h_{x=y} + h_{y=z} - h_{x=z} \leq 1 \tag{3.3}$$

In Figure 3.1, the constraint $h_{x=y} + h_{A=x} - h_{y=A} \leq 1$ is generated as an instance of inequality (3.3).

Constraint 3: From the definition of h variables, h_p must be 1 if there exists set Q of literals that implies p are explicitly or implicitly hypothesized (i.e. for all $Q \subseteq P$, $(Q \cup B \models p \wedge H \cup B \models Q) \Rightarrow H \cup B \models p$). We introduce new ILP variable $a_Q \in \{0, 1\}$ for set Q of elemental explanations s.t. $a_Q = 1$ iff all literals in Q is a logical consequence of $H \cup B$; $a_Q = 0$ otherwise. Using a_Q, the constraint $\forall Q \subseteq P[(H \cup B \models Q \wedge Q \cup B \models p) \Rightarrow h_p = 1]$ can be expressed as follows:

$$\text{for each } p \in P : \sum_{Q \in \mathcal{E}(p)} a_Q \leq |\mathcal{E}(p)| \cdot h_p, \tag{3.4}$$

where $\mathcal{E}(p)$ is a set of set of potential elemental explanations that explain p. For example, in Figure 3.1, the constraint $a_{\{s(y),t(u)\}} \leq h_{s(x)}$ is generated.

Constraint 4: From the definition of an ILP variable a, for all $Q \subseteq P$, a_Q can be set to 1 if and only if Q is a logical consequence of $H \cup B$ (for all $q \in Q$,

$h_q = 1$). This can be expressed as follows:

$$\text{for each } p \in P, Q \in \mathcal{E}(p) : |Q| a_Q \leq \sum_{q \in Q} h_q \tag{3.5}$$

$$\text{for each } p \in P, Q \in \mathcal{E}(p) : \sum_{q \in Q} h_q \leq |Q| - 1 + a_Q \tag{3.6}$$

In this formulation, we generate $O(n^3)$ ILP constraints for Constraint 2, where n is the number of logical atomic terms appearing in P. As the reader will see in Sec. 4.4, this makes inference intractable in large-scale processing. We propose how this drawback can be overcome by exploiting Cutting Plane Inference in Sec. 3.3.2.

3.2.2 Formulation for Implementing The Cost Function

As mentioned in Sec. 2.3, we adopt the cost function proposed by Hobbs et al. (1993). For convenience, we repeat the cost function:

$$cost(H) = \sum_{h \in A(H)} cost(h), \tag{3.7}$$

where $A(H)$ is a set of *assumed* literals in H. This means that the cost of H is calculated from the subset of hypothesized literals. To represent the set of literals that are counted in the cost function, we first introduce ILP variables $c \in \{0, 1\}$ as follows:

$$\text{for each } p \in P : c_p = \begin{cases} 1 & \text{if } p \text{ pays its cost} \\ 0 & \text{otherwise.} \end{cases}$$

In Figure 3.1, $c_{s(x)}$ will be set to 0 in H_2 since $s(x)$ does not pay the cost (i.e. $s(x)$ is explained by $r(x)$).

Using c variables, the objective function of the ILP problem is given by:

$$\text{minimize } cost(H) = \sum_{p \in P} c_p \cdot cost(p) \tag{3.8}$$

Note that it is easy to incorporate another criteria into the cost function. For

instance, one can consider the plausibility of coreference relation between two mentions in a text. Assuming mentions are represented by variables (e.g. $cat(x)$ means that the mention x whose linguistic expression is *cat* appears in a text), one can add $\sum_{x,y \in T} cost(x, y, O) \cdot h_{x=y}$, where the cost is calculated by the information mentioned in O. For example, one could design the cost function that returns a higher cost if two contradictory properties are mentioned in O (e.g $cat(x)$ and $dog(y)$ occur in O).

Again, from the definition of c variables, not all value assignments to c are allowed. Accordingly, we introduce several constraints on c as follows.

Constraint 5: From the definition of the cost function in Sec. 2.3, c_p is set to 1 if and only if (i) p is *not* explained (i.e. assumed), and (ii) p is *not* unified with any other literal that has the smaller cost by factoring of H. To represent the second case, we introduce new ILP variable $u_{p_1,p_2} \in \{0, 1\}$ for the pair (p_1, p_2) of unifiable literals s.t. $u_{p_1,p_2} = 1$ iff p_1 is unified with p_2 by factoring of H; $u_{p_1,p_2} = 0$ otherwise. Using u, the condition can be expressed as follows:

$$\text{for each } p \in P: \quad h_p + \sum_{Q \in \mathcal{E}(p)} (1 - a_Q) + \sum_{p' \in U^-(p)} (1 - u_{p,p'}) \leq |\mathcal{E}(p)| + |U^-(p)| + c_p \tag{3.9}$$

$$\text{for each } p \in P: \quad (1 + |\mathcal{E}(p)| + |U^-(p)|) \cdot c_p \leq h_p + \sum_{Q \in \mathcal{E}(p)} (1 - a_Q) + \sum_{p' \in U^-(p)} (1 - u_{p,p'}), \tag{3.10}$$

where $U^-(p)$ is a set of literals that (i) are unifiable with p, and (ii) have the cost smaller than $cost(p)$. For example, in Figure 3.1, we introduce $h_{s(x)} + (1 - a_{\{r(x)\}}) + (1 - u_{s(x),s(y)}) \leq 2 + c_{s(x)}$ as an instance of inequality (3.9).

Finally, we impose a constraint on u_{p_1,p_2} so that the value of u_{p_1,p_2} is allowed to be 1 only if (i) there exists the equalities that make p_1 and p_2 equivalent in H, and (ii) p_1 and p_2 are hypothesized.

Constraint 6: By the definition of an ILP variable u, $u_{p_1(\mathbf{x}),p_2(\mathbf{y})}$ can be set to 1 if and only if (i) two literals $p_1(\mathbf{x}) \equiv p_1(x_1, x_2, ..., x_n)$ and $p_2(\mathbf{y}) \equiv$

$p_2(y_1, y_2, ..., y_n)$ are unified (i.e. the substitution $\{x_i/y_i\}_{i=1}^n$ occurs, namely $h_{x_i=y_i} = 1$ for all $i \in \{1, 2, ..., n\}$), and (ii) both $p_1(\mathbf{x})$ and $p_2(\mathbf{x})$ are hypothesized.

$$(n+2) \cdot u_{p_1(\mathbf{x}),p_2(\mathbf{y})} \leq \sum_{i=1}^{n} h_{x_i=y_i} + h_{p_1(\mathbf{x})} + h_{p_2(\mathbf{y})} \tag{3.11}$$

$$\sum_{i=1}^{n} h_{x_i=y_i} + h_{p_1(\mathbf{x})} + h_{p_2(\mathbf{y})} \leq (n+2) - 1 + u_{p_1(\mathbf{x}),p_2(\mathbf{y})} \tag{3.12}$$

In Figure 3.1, the constraint $(1+2) \cdot u_{r(x),r(A)} \leq h_{x=A} + h_{r(x)} + h_{r(A)}$ is generated. Finally, in order to avoid the case where we hypothesize a single literal that (i) does not explain anything, but (ii) is unified with the other literal, we impose the following constraint:

$$\text{for each } p \in P: \quad h_p \leq \sum_{Q \in C(p)} a_Q, \tag{3.13}$$

where $C(p)$ is a set of set of literals with which p co-occur to explain the other literal. In Figure 2, since $s(y)$ co-occurs with $t(u)$ to explain $q(y)$ (i.e. $C(s(y)) = \{\{s(y), t(u)\}\}$), we introduce $h_{s(y)} \leq a_{s(y),t(u)}$. If we do not have this constraint, we can hypothesize $s(y)$ without hypothesizing $t(u)$ to reduce the cost of $s(x)$. Since such a hypothesis cannot be generated through backward-inference, we need to prohibit it.

We show the soundness and completeness of the proposed formulation in Sec. 3.3.1, showing how to handle negation in our formulation.

3.3 Improving Expressiveness and Efficiency of ILP Formulation

The presented ILP formulation is still imperfect in terms of the expressivity and efficiency. The first problem is that it does not support negative literals in either background knowledge, observation or explanation. The lack of support of negation does not allow us to represent a negative proposition, which is often required

in the discourse processing problems. We thus introduce two formulations for making the ILP formulation support negation (Sec. 3.3.1). The second problem is that we need to generate $O(n^3)$ transitivity constraints, where n is the number of logical atomic terms (see Constraint 2). This often makes inference intractable in large-scale inference. We improve the inefficiency by employing Cutting Plane Inference, which is an iterative optimization strategy developed in Operations Research (Sec. 3.3.2).

3.3.1 Handling Negation in ILP-based Formulation

The capability of handling negations is crucial for a wide range of abductive inference systems. For example, in abduction-based natural language interpretation, one can easily imagine that it needs to handle negated expressions, such as "*I don't like ice cream.*", or "*Tweety is not a bird.*", etc. Traditionally, there are two big paradigms of negation implementation in the context of logic programming, where negation operator is treated under two different semantics: (i) classical negation, and (ii) negation as failure. For classical negation, the negation operator is interpreted as negation in classical logic (i.e. $\neg p$ means that the proposition p is false). Negation as failure is based on *closed world assumption* (CWA) [Reiter, 1978], which assumes that background knowledge represents all true facts, and propositions that cannot be proven are concluded to be false. Therefore, $\neg p$ means that p is false *if* we cannot prove p. Many logic programming software, e.g. Prolog, adopts negation as failure, where the negation operator is written as *not* for the clarity of different semantics.

In this section, we show how to implement negation in the ILP formulation under the semantics of classical negation. That is, by "hypothesize $\neg p$", we mean that we assume that p is false. It does *not* mean that we assume that p is false *if* p cannot be proven. The latter semantics could be adopted in analogy to SLDNF-resolution [Apt and van Emden, 1982], which is an extension of SLD-resolution [Robinson, 1965] with negation as failure support.

In SLDNF-resolution, negative literal $\neg p$ in the goal expression invokes additional SLD-resolution with the goal $\leftarrow p$ to decide whether $\neg p$ is true or not. Applying this idea to our framework, one could use the explanation of p as the

explanation of $\neg p$, where the cost of $\neg p$ is inverse proportion to the cost of explanation of p (i.e. the better p is explained, the less probable $\neg p$ being true is). We will purse this direction in future work.

Let us redefine the formats of background knowledge, observation, and explanation as follows:

- Background knowledge: a set of first-order logical formulae in the form $l_1 \wedge l_2 \wedge ... \wedge l_n \rightarrow m$, where $l_1, l_2, ..., l_n$, and m are literals;
- Observations: a set of literals or equalities;
- Explanation: a set of literals or equalities.

In the rest of this section, we give two formulations for expressing negative literals and inequality of variables (i.e. $x \neq y$) for the framework described in Sec. 3.1. Three non-trivial questions arise when the ILP-based framework supports negation: (i) how to represent logical negation in terms of ILP variables, (ii) how to exclude inconsistent explanations from the search space of candidate explanations, and (iii) whether the extended formulation is sound and complete; that is, the search space represented by the extended formulation does not include inconsistent explanations, and none of valid candidate explanations are excluded from the search space.

First, consider the case where there are two literals $p(x_1, x_2, ..., x_n) = p(\mathbf{x})$ and $\neg p(y_1, y_2, ..., y_n) = \neg p(\mathbf{y})$ in set P of potential elemental explanations such that $p(\mathbf{x})$ and $p(\mathbf{y})$ are unifiable. Concerning the issue (i), we represent negative literal $\neg a$ as $h_{\neg a}$ in the ILP optimization problem. Recall that $h_a = 0$ does not mean $\neg a$. To address the issue (ii), we want to prohibit $H \cup B \models p(\mathbf{x}) \wedge \neg p(\mathbf{x})$, namely prevent the two literals from being hypothesized simultaneously (i.e. $h_{p(\mathbf{x})} = 1$ and $h_{p(\mathbf{y})} = 1$) if $\mathbf{x} = \mathbf{y}$ is implied by $H \cup B$ (i.e. $h_{x_i = y_i} = 1$ for all $i \in \{1, 2, ..., n\}$). Therefore we introduce the following constraint.

Constraint 7: By the definition of explanation in Sec. 2.3 ($H \cup B \not\models \perp$), two contradictory literals $p(x_1, x_2, ..., x_n) \equiv p(\mathbf{x})$ and $\neg p(y_1, y_2, ..., y_n) \equiv \neg p(\mathbf{y})$ cannot be both hypothesized ($h_{p(\mathbf{x})} = 1$ and $h_{\neg p(\mathbf{y})} = 1$) if $x_i = y_i$ are

hypothesized ($h_{x_i=y_i} = 1$) for all $i \in \{1, 2, ..., n\}$. This can be expressed as:

$$h_{p(\mathbf{x})} + h_{\neg p(\mathbf{y})} + \sum_{i=1}^{n} h_{x_i=y_i} \leq 1 + n. \tag{3.14}$$

Notice that the case where $\mathbf{x} = \mathbf{y}$ reduces to: $h_{p(\mathbf{x})} + h_{\neg p(\mathbf{x})} \leq 1$. This type of constraint grows in $O(nm)$ for each predicate p, where n is the number of positive instantiation of p in P, and m is the number of negative instantiation of p in P.

Inequalities can also be formulated as the special case of inequality (3.14). Similarly to negative literals, we represent $x \neq y$ as $h_{x \neq y}$ in the ILP optimization problem. We then prohibit two contradictory equalities $x = y$ and $x \neq y$ from being hypothesized simultaneously:

$$h_{x=y} + h_{x \neq y} + \leq 1 \tag{3.15}$$

The important question here is how to find the pairs of potentially contradictory literals (i.e. $p(\mathbf{x})$ and $\neg p(\mathbf{y})$). Algorithm 1 does not enumerate all the set of literals that can be logical consequences of $H \cup B$, because the algorithm uses backward-inference for creating set P of potential elemental explanations. As a result, the system loses its soundness: an inconsistent explanation can be chosen as the best explanation. For example, given $O = \{p(a), \neg r(a)\}$ and $B = \{q(x) \rightarrow p(x), q(x) \rightarrow r(x)\}$, Algorithm 1 generates $P = \{p(a), \neg r(a), q(a)\}$. Let us consider one inconsistent explanation $H = \{q(a), \neg r(a)\}$. Although $r(a)$ is a logical consequence of $H \cup B$, Algorithm 1 does not generate the ILP variable $h_{r(a)}$ and the ILP constraint $h_{r(a)} + h_{\neg r(a)} \leq 1$. Therefore, the system can incorrectly output the inconsistent explanation $H = \{q(a), \neg r(a)\}$ as the best explanation.

In order to avoid this problem, one could exploit *deduction* of $B \cup P$ to let P include all the possible logical consequences of $H \cup B$. The clause C is said to be *deduced* from the set Σ of clauses iff there exists the clause D such that D can be derived from Σ through resolution, and D subsumes C. Lee (1967) showed the completeness of deduction: every logical consequence of Σ can be deduced from Σ,

where Σ is a set of logical formulae (*called* Subsumption theorem). Therefore, one could use deduction to add a complete set of logical consequences of $H \cup B$ that are potentially implied by $H \cup B$, and add them to the search space. As shown below, the system is proven to be sound and complete over such a search space. In practical, as a deductive inference system, one could use many sophisticated deductive inference engines that have been developed so far. In our experiments in Sec. 4.4, however, we use Constraint 6 *without* performing deduction, and the empirical evaluation with deduction is our future work. Finally, to show that our system is sound and complete, we prove the following theorem:

Theorem 3.3.1 Let B be a background knowledge, O be observations, and P be a set of potential elemental explanations w.r.t. B and O. Suppose P is constructed in the following way: (i) we first execute the procedure *liftedFirstOrderCBA* in Algorithm 1, and (ii) we add a set of literals derived by deduction from $P \cup B$ to P. Let S_H be a 0-1 value assignment function to ILP variables h (i.e. $P \rightarrow \{0, 1\}$) introduced in Sec. 3.1. Let H_S be a candidate explanation, which is constructed from S_H followed by the definition 3.2.1, namely $H_S = \{p \mid p \in P, S_H(p) = 1\}$. Then, the following proposition is true:

- H_S is a candidate explanation if and only if S_H is a solution of ILP optimization problem (3.7), namely S_H satisfies the whole constraints introduced in Sec. 3.1 and Sec. 3.3.1.

Proof See Appendix 7.2.6.

3.3.2 Cutting Plane Inference for CBA

One major drawback of the ILP formulation is that it needs to generate $O(n^3)$ transitivity constraints, where n is the number of logical atomic terms, because we perform inference over FOL-based representation. That makes inference intractable (see Sec. 4.4 for empirical evidence) because it generates an ILP optimization problem that has quite a large number of constraints. Moreover, handling negation quadratically increases Constraint 7 described in Sec. 3.3.1.

How do we overcome this drawback? The idea is that "all the transitivity constraints may not be violated all at once; so we gradually optimize and add

Algorithm 3 cpiForliftedFirstOrderCBA(Background Knowledge **B**, Observation **O**)

1: $(\Psi, I) \leftarrow createBaseILP(B, O)$
2: **repeat**
3: $sol \leftarrow solveILP(\Psi, I)$; $V \leftarrow \{\}$
4: **for** $x, y \in \{t_1, t_2 \mid t_1 \in T, t_2 \in T, sol(h_{t_1=t_2}) = 1)$ **do**
5: **for** $z \in termsUnifiableWith(x, y)$ **do**
6: // $H \cup B \models x = y \wedge y = z$, and $H \cup B \not\models x = z$
7: **if** $sol(h_{y=z}) = 1$ and $sol(h_{x=z}) = 0$ **then** $V \leftarrow V \cup \{h_{x=y} + h_{y=z} - h_{x=z} \leq 1\}$
8: // $H \cup B \models x = y \wedge x = z$, and $H \cup B \not\models y = z$
9: **if** $sol(h_{y=z}) = 0$ and $sol(h_{x=z}) = 1$ **then** $V \leftarrow V \cup \{h_{x=y} + h_{x=z} - h_{y=z} \leq 1\}$
10: **end for**
11: **end for**
12: $I \leftarrow I \cup V$
13: **until** $V \neq \phi$

transitivity constraints *if violated* in an iterative manner." More formally, we propose to apply *Cutting Plane Inference (CPI)* to the CBA problems. CPI is an exact inference optimization technique that is originally developed for solving large linear programming (LP) problems in Operations Research [Dantzig et al., 1954]. CPI has been successfully applied to a wide range of constrained optimization problems where constraints are very large [J. Berant and Goldberger, 2008; Riedel, 2008; Riedel and Clarke, 2006; T. Joachims, 2009, etc.], from probabilistic deductive inference problems [Riedel, 2008] to machine learning problems [T. Joachims, 2009]. To the best of our knowledge, however, our work is the first successful work to apply CPI to abductive inference tasks. In principle, CPI solves optimization problem in an iterative manner as follows: it solves an optimization problem without constraints, and then adds violated constraints to the optimization problem. When the iteration terminates, it guarantees solutions to be optimal. The proposed algorithm, called *CPI4CBA*, is also an exact inference framework.

How do we apply the technique of CPI to cost-based abduction problems? Intuitively, we iterate the following two steps: (i) solving an abduction problem without enforcing transitivity on logical atomic terms, and (ii) generating transi-

tivity constraints dynamically when transitiveness of unification is violated (e.g. $H \cup B \models x = y \wedge y = z$, and $H \cup B \not\models x = z$). The iteration terminates if there is no violated unification transitivity. The pseudo-code is given in Algorithm 3. In line 1, we first create an ILP optimization problem described in Sec. 3.1 and Sec. 3.3.1 but without transitivity constraints (i.e. Constraint 2), where Ψ denotes a set of ILP variables, and I denotes a set of ILP constraints. In line 2–13, we repeat: checking consistency of unification transitiveness, adding constraints for violated transitiveness, and re-optimizing. In line 3, we find the solution *sol* for the current ILP optimization problem. Then, for each pair (x, y) of logical atomic terms unified in the solution *sol* (line 4), find the logical term z which is unifiable with x and y (line 5). If the transitive relation x, y with respect to z is violated (i.e. $h_{x=z} = 0 \wedge h_{y=z} = 1$ or $h_{x=z} = 1 \wedge h_{y=z} = 0$), then we generate constraints for preventing this violation, and keep it in set V of constraints (line 6–9). Finally, we again perform an ILP optimization with newly generated constraints (line 12 and 3). The iteration ends when there is no violated transitiveness (line 13).

The key advantages of CPI4CBA is that it can reduce the time of search-space generation, and it is also expected to reduce the time of ILP optimization. CPI4CBA does not generate all the transitivity constraints before optimization, which saves the time for search-space generation. In addition, optimization problems that we solve would become smaller than the original problem in most cases, because not all the transitivity constraints may not be necessary to be considered. In the worst case, we need to solve the optimization problem that is same as the original one; but in most cases we found out that we do not need to. We will show its empirical evidence through large-scale evaluation in Sec. 4.4.

3.4 Runtime Evaluation

How much does CPI improve the *runtime* of ILP-based reasoner? Does CPI scale to larger real-life problems? To answer these questions, we evaluated the CPI4CBA algorithm in two settings: (i) **STORY**, the task of plan recognition, and (ii) **RTE**, the popular, knowledge-intensive, real-life natural language processing task of *Recognizing Textual Entailment* (RTE). While most of the ex-

isting abductive inference systems are evaluated on rather small, and/or artificial datasets [Kate and Mooney, 2009; Raghavan and Mooney, 2010; Singla and Domingos, 2011, etc.], our evaluation takes a real-life, much larger datasets (see Sec. 3.4.1). In our experiments, we compare our system with the systems [Blythe et al., 2011; Kate and Mooney, 2009; Singla and Domingos, 2011] based on Markov Logic Networks (MLNs) [Richardson and Domingos, 2006]. For our experiments, we have used a 12-Core Opteron 6174 (2.2GHz) 128 GB RAM machine, and assigned 8 cpu cores for each run. For an ILP solver, we used Gurobi Optimizer.[1] It is commercial but an academic license is freely available.

3.4.1 Settings

STORY: For this setting, we have used Ng and Mooney (92)'s story understanding dataset,[2] which is widely used for evaluation of abductive plan recognition systems [Kate and Mooney, 2009; Raghavan and Mooney, 2010; Singla and Domingos, 2011]. In this task, we need to abductively infer the top-level plans of characters from actions. We follow Singla and Mooney's setting to define top-level plan predicates. The top-level plan predicates include 10 types of literals, such as *shopping*.[3] The dataset consists of 50 plan recognition problems represented by a set of ground atoms (e.g. $\{$*getting_off*(*Getoff16*), *agent_get_off*(*Getoff16*, *Fred16*) *name*(*Fred16*, *Fred*)$\}$) and 107 background Horn clauses (e.g. $go\_step(r, g) \wedge going(g) \rightarrow robbing(r)$). The dataset contains on average 12.6 literals in the logical forms of actions. To make the predicates representing top-level plans (e.g. shopping, robbing) disjoint, we generated 73 disjointness axioms by using the formulation described in Sec. 3.3.1.[4]

To assign a cost to each literal (i.e. $cost(h)$ in the equation (3.8)), we followed Hobbs et al. (1993)'s weighted abduction theory. In the theory, as mentioned in Sec. 2.3, each literal in the left-hand side of axioms has a set of *weights*, which is expressed as $p_1^{w_1} \wedge p_2^{w_2} \wedge ... \wedge p_n^{w_n} \rightarrow q$. During backward-chaining, each weight

[1] http://www.gurobi.com/

[2] ftp://ftp.cs.utexas.edu/pub/mooney/accel

[3] The complete list of top-level plan predicates: shopping, robbing, traveling, rest_dining, drinking, paying, jogging, and partying.

[4] For example, $robbing(x) \wedge shopping(x) \rightarrow \bot$ is represented by $h_{robbing(x)} + h_{shopping(y)} + h_{x=y} \leq 2$.

is multiplied with the cost of literal that is backchained on. For example, given $p(x)^{0.6} \wedge q(x)^{0.6} \rightarrow r(x)$ and $r(a)^{\$10}$, the theory derives $\{p(a)^{\$6}, q(a)^{\$6}\}$. Because the background knowledge of Ng and Mooney (92)'s dataset does not have weights, we assigned the weights to axioms so that the sum of the weights is 1.2 (e.g. $p^{0.4} \wedge q^{0.4} \wedge r^{0.4} \rightarrow s$). This assignment means that backward-inference always increases the cost of explanation, and unification is the only way to reduce the cost. That is, it is almost equivalent to performing pure logic-based abduction, where the number of literals in an explanation is used as the plausibility of explanation.

RTE: For observations (input), we employed the second challenge of RTE dataset.[1] In the task of RTE, we need to correctly determine whether one text (called *text*, or T) entails another (called *hypothesis*, or H) or not. The dataset consists of development set and test set, each of which includes 800 natural language text-hypothesis pairs. We have used all of the 800 texts from test set. We have converted texts into logical forms presented in [Hobbs, 1985] using the Boxer semantic parser [Bos, 2008]. The number of literals in observations is 29.6 literals on average. For background knowledge, we have extracted 289,655 axioms[2] from WordNet 3.0 [Fellbaum, 1998], and 7,558 axioms from FrameNet 1.5 [Ruppenhofer et al., 2010] following Ovchinnikova et al. [2011]. In principle, the WordNet knowledge base contains several kinds of lexical relations between words, such as IS-A, ontological relations (e.g. $dog(x) \rightarrow animal(x)$). FrameNet knowledge bases contain lexeme-to-frame mappings, frame-frame relations, etc. For example, the mapping from surface realization "give to" to a frame "Giving" is given by: $Giving(e_1, x_1, x_2, x_3) \wedge donor(e_1, x_1) \wedge recipient(e_1, x_2) \wedge theme(e_1, x_3) \rightarrow give(e_1, x_1, x_3) \wedge to(e_2, e_1, x_2)$. We again followed Hobbs et al. [1993]'s weighted abduction theory for calculating the cost of explanation. We assigned the weights to axioms by following Ovchinnikova et al. [2011] in this setting.

3.4.2 Results and Discussion

The reasoner was given a 2-minute time limit for each inference step (i.e.

[1] http://pascallin.ecs.soton.ac.uk/Challenges/RTE2/

[2] Extracted relations are: word-to-synset mapping, hypernym-hyponym, cause-effect, entailment, derivational, instance-of relations.

Setting	Method	Depth	Generation [sec.] (timeout = 120)	ILP inf [sec.] (timeout = 120)	# of ILP cnstr
STORY	IAICBA	1	0.02 (100.0 %)	0.60 (100.0 %)	3,708
		2	0.12 (100.0 %)	5.34 (100.0 %)	23,543
		3	0.33 (100.0 %)	8.11 (100.0 %)	50,667
		∞	0.35 (100.0 %)	9.00 (100.0 %)	61,122
	CPI4CBA	1	0.01 (100.0 %)	0.34 (100.0 %)	784 (Δ 451)
		2	0.07 (100.0 %)	4.15 (100.0 %)	7,393 (Δ 922)
		3	0.16 (100.0 %)	3.36 (100.0 %)	16,959 (Δ 495)
		∞	**0.22 (100.0 %)**	**5.95 (100.0 %)**	24,759 (Δ 522)
RTE	IAICBA	1	0.01 (100.0 %)	0.25 (99.7 %)	1,104
		2	0.08 (100.0 %)	2.15 (98.1 %)	5,185
		3	0.56 (99.9 %)	5.66 (93.0 %)	16,992
		∞	4.78 (90.7 %)	15.40 (60.7 %)	36,773
	CPI4CBA	1	0.01 (100.0 %)	0.05 (100.0 %)	269 (Δ 62)
		2	0.04 (100.0 %)	0.35 (99.6 %)	1,228 (Δ 151)
		3	0.09 (100.0 %)	1.66 (99.0 %)	2,705 (Δ 216)
		∞	**0.84 (98.4 %)**	**11.73 (76.9 %)**	10,060 (Δ 137)

Table 3.1: The results of averaged inference time in **STORY** and **RTE**.

search-space generation and best-explanation search). In Table 3.1, we show the results of each setting for two inference method in Table 3.1: (i) *IAICBA*: the inference method without CPI, and (ii) *CPI4CBA*: inference method with CPI. In order to investigate the relation between the size of search space and the runtime, we show the results for each depth, which we used for limiting the length of backward-chaining. In the "Generation" column, we show the runtime that is taken for search-space generation in seconds averaged over all problems whose search-space generation is finished within 2 minutes. In the parenthesis, we show the percentage of those problems whose search-space generation is finished within 2 minutes. In the column "ILP inf", we show the runtime of ILP optimization averaged on only problems such that both search-space generation and ILP optimization are finished within 2 minutes, as well as the percentage of those problems (e.g. 80 % means "for 80 % of all the problems, search-space generation was finished within 2 minutes, and so was ILP inference."). In the "# of ILP cnstr" column, we show the averaged number of generated ILP constraints. Concerning

CPI4CBA, the number denotes the averaged number of constraints considered in the end, including the constraints added by CPI. The number marked by Δ indicates the averaged number of constraints that are added during CPI (i.e. how many times are the constraints added by line 7 or 9 in Algorithm 3).

Overall, the runtimes in both search-space generation and ILP inference are dramatically improved from IAICBA to CPI4CBA in both settings, as shown in Table 3.1. In addition, CPI4CBA can find optimal solutions in ILP inference for more than 90 % of the problems, even for depth ∞. This indicates that CPI4CBA scales to larger problems. From the results of IAICBA in **RTE** settings, we can see the significant bottleneck of IAICBA in large-scale reasoning: the time of search-space generation. The search-space generation could be done within 2 minutes for only 90.7 % of the problems. CPI4CBA successfully overcomes this bottleneck. CPI4CBA is clearly advantageous in the search-space generation because it is not necessary to generate transitivity constraints, an operation that grows cubically before optimization.

In addition, CPI4CBA also reduces the time of ILP inference significantly. In ILP inference, CPI did not guarantee the reduction of inference time in theory; *however*, as shown in Table 3.1, we found that the number of ILP constraints actually used is much less than the original problem. Therefore, CPI4CBA successfully reduces the complexity of the ILP optimization problems in practice. This is also supported by the fact that CPI4CBA keeps 93.9% in "ILP inf" for Depth $= \infty$ because it solves very large ILP optimization problems that fail to be generated in IAICBA. In order to see how CPI contributes to the improvement of ILP inference time, we show how the runtime of IAICBA is affected by CPI4CBA method for each problem in Figure 3.2. Each data point corresponds to one problem in **STORY** and **RTE** settings. We show the data points for problems that we found optimal solutions in ILP inference for Depth $= \infty$. Overall, the runtime of CPI4CBA is smaller than IAICBA in most problems. In particular, we can see that CPI4CBA successfully reduces the time of ILP inference for larger problems by exploiting the iterative optimization technique. In the larger domain of **RTE** setting, we found that the performance was improved in 81.7 % of the problems.

Finally, we compare our results with other existing systems. Regarding the MLN-based systems [Blythe et al., 2011; Kate and Mooney, 2009; Singla and

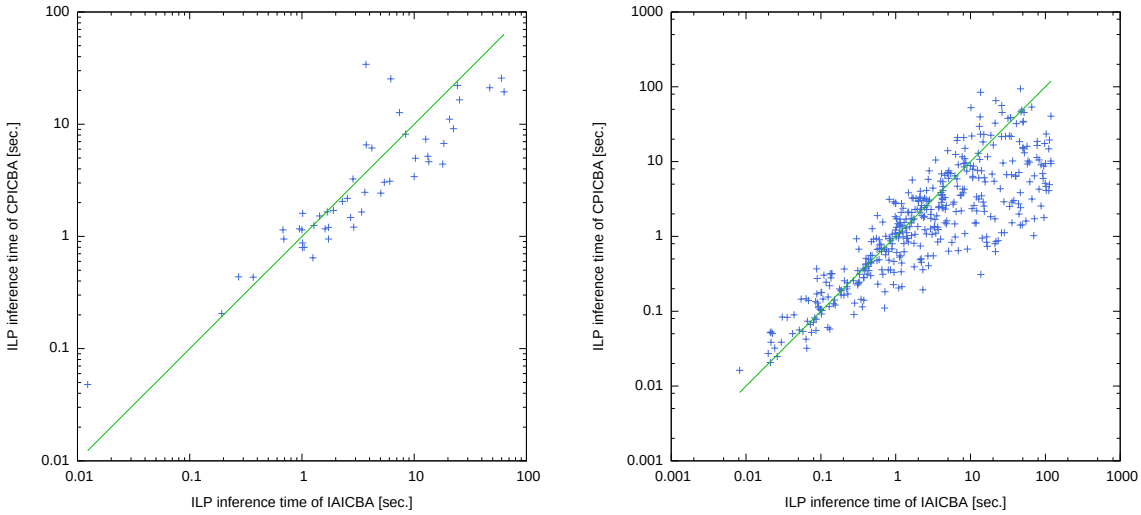


Figure 3.2: Runtime comparison between IAICBA and CPI4CBA (logarithmic scale). The left figure shows the results of STORY dataset, and the right figure shows the results of RTE datasets.

Domingos, 2011], our results are comparable or slightly less efficient in the **STORY** setting, and more efficient than the existing systems in the **RTE** setting. For the **STORY** setting, Singla and Mooney (2011) report the results of two systems with an exact inference technique using CPI for MLNs [Riedel, 2008]: (i) Kate and Mooney (2009)'s approach: 2.93 seconds, and (ii) Singla and Mooney (2011)'s approach: 0.93 seconds.[1] To make the comparison fair, we evaluated our approach with one CPU core. The inference time is 31.3 seconds on average (optimal solutions were found for the 80 % of the problems). MLN-based approaches seem to be reasonably efficient for small datasets. However, it does not scale to larger problems; for the **RTE** setting, Blythe et al. (2011) report that only 28 from 100 selected RTE-2 problems could be run to completion with only the FrameNet knowledge bases. The processing time was 7.5 minutes on average (personal communication).[2] On the other hand, our method solves 76.9% of all the problems, where suboptimal solutions are still available for the rest of 21.5%,

[1]This is the result of MLN-HC in Singla and Domingos [2011]. MLN-HCAM cannot be directly compared with our results, since the search space is different from our experiments because they unify some assumptions in advance to reduce the search space.

[2]They used 56,000 FrameNet axioms in the experiments, while we used 289,655 WordNet axioms and 7,558 FrameNet axioms.

and it takes only 0.84 seconds for search-space generation, and 11.73 seconds for ILP inference. As mentioned in Sec. 4.5, our framework is more scalable because our framework does not need to explicitly generate the axioms to emulate *explaining away* effect (i.e. inferring one cause makes another cause less probable), and need no grounding.

3.5 Related Work

The computational aspect of abduction has been studied extensively in the contexts of logic programming and Statistical Relational Learning. In the context of logic programming, abduction has been introduced as the extension of logic programming [Kakas et al., 1992; Stickel, 1991, etc.], where the extended framework is often called Abductive Logic Programming (ALP). Since abduction and induction share the basic framework (see Sec. 2.3 for detail), abduction has also been studied in the area of Inductive Logic Programming, the logic programming framework for induction [Inoue, 2004; Tamaddoni-Nezhad et al., 2006, etc.]. In the context of ALP, Stickel (1991) showed how to formulate minimum-cost explanation finding in Prolog, a popular implementation of logic programming. Stickel allows the system to *assume* literals during the SLD resolution when Horn-clause rules or facts unifiable with the targeted literal are not found. In their system, the cost of explanation is calculated by the sum of the costs of elemental explanations, and the costs of axioms used for constructing the proof. However, Stickel does not show how to implement it in a efficient way. After a few years, a number of methods attempting to efficiently find the minimum-cost explanation have been proposed [Abdelbar and Hefny, 2005; Chivers et al., 2007; Guinn et al., 2008; Ishizuka and Matsuo, 1998; Prendinger and Ishizuka, 1999; Santos, 1994, etc.]; for example, Santos (1994) formulated cost-based abduction in propositional logic using ILP, and showed its efficiency. However, most of them focus on improving the inefficiency of propositional logic-based abduction. As discussed in Sec. 3.1, one could use such a framework through propositionalization techniques for first-order CBA; however, the propositionalization will produce a huge amount of ground instances of background knowledge axioms and literals in observation. Hence they would not scale to larger problems with large knowledge bases.

3.5.1 Comparison with Santos's ILP-based Formulation

The most similar previous work to us is Santos (1994)'s ILP-based formulation of propositional logic-based CBA. Our approach is different from Santos (1994)'s LP formulation in two ways. The first difference is that we are capable of evaluating the specificity of explanations, which is one of important features for abduction-based NLP as discussed in Sec. 2.3. Santos's approach amounts to performing most-specific abduction, and they find a truth assignment to all the propositions in the world. Let us describe how the appropriate level of specificity is controlled in our approach. Suppose $O = \{p(a), q(a)\}$, and $B = \{r(x) \rightarrow p(x)\}$. We then have two candidate explanations. The first explanation is $H_1 = \{p(a), q(a)\}$, which simply assumes observations, and the cost is $cost(p(a)) + cost(q(a))$ (i.e. $c_{p(a)} = 1$, $c_{q(a)} = 1$). Backward-chaining on $p(a)$ yields the second explanation $H_2 = \{q(a), r(a)\}$, which is more specific than H_1. The cost of H_2 is $cost(q(a)) + cost(r(a))$ ($c_{p(a)} = 0$, $c_{q(a)} = 1$, $c_{r(a)} = 1$). Note that we do not count $p(a)$ because $p(a)$ is *not* assumed anymore. Therefore, for this problem, if $cost(r(a)) < cost(p(a))$, then more specific explanation H_1 is selected as the best explanation; otherwise, the less specific explanation H_2 is selected as the best explanation. This is controlled by the ILP variables c and Constraints 5 and 6, which are not introduced in Santos (1994)'s approach. To summarize, our approach can decide which specificity of explanation is appropriate for the current observation and knowledge base, based on how well the explanation is supported by observations.

Another difference from Santos (1994)'s approach is that our approach directly models first-order CBA, while his approach formulates propositional-logic abduction. We could employ their approach for first-order CBA since it is well-known that FOL formulae can be represented by propositional logic formulae through the application of grounding procedure (i.e. generate logical formulae, replacing variables with all possible constants). However, abductive inference over propositional level will make inference intractable when existentially quantified variables are included in observations or background knowledge. For example, suppose that $B = \{q(x, y) \rightarrow p(x, y), r(x, y) \rightarrow \exists z q(x, z)\}, O = \{p(x, y)\}$ and all possible constants are $\mathcal{C} = \{C_1, C_2, ..., C_n\}$. To ground this observation, we need to generate a disjunctive clause for $p(x, y)$, replacing x and y with all possible

combinations from $\mathcal{C}$, i.e. $p(C_1, C_1) \vee p(C_1, C_2) \vee ... \vee p(C_n, C_n)$. Extending the expressivity of observation is not a difficult work, but the problem is: in the search-space generation process, we get $O(n^2)$ potential elemental explanations (i.e. $q(C_i, C_j)$ for all $i, j \in \{1, 2, ..., n\}$) to explain each disjunct. In addition, backchaining on each $q(C_i, C_j)$ with $r(x, y) \rightarrow \exists z q(x, z)$ yields $O(n)$ potential elemental explanations (i.e. $r(C_i, C_k)$ for all $k \in \{1, 2, ..., n\}$). In contrast, the search-space generation in our approach yields $\{p(x, y), q(x, y), r(z, u)\}$. As the readers can see, our approach seems to be more robust to the size of domain.

3.5.2 Comparison with Other Logic-based Formalisms

In the context of Statistical Relational Learning, abduction has also been widely studied. One of the prominent formalisms is PRISM [Sato and Kameya, 2008], which is a general logic-based probabilistic modeling language. In the past two decades, a number of the techniques for efficient inference or learning has been studied extensively (see [Sato and Kameya, 2008] for overview). Concerning inference, in principle, PRISM achieves the best explanation finding in a polynomial time through a tabled search technique for logic programs [Tamaki and Sato, 1986]. However, the tabled search technique exploits local information that is computed so far, and hence is incompatible with the factoring of explanation, which is rather global operation (personal communication). It is non-trivial issue to incorporate the factoring process into the search without the loss of efficiency.

Another important stream is the series of studies [Blythe et al., 2011; Kate and Mooney, 2009; Singla and Domingos, 2011, etc.], where abduction has been emulated through Markov Logic Networks (MLNs) [Richardson and Domingos, 2006], a probabilistic deductive inference framework. MLNs provide full support of first-order predicate logic and the software packages of inference and learning; however, MLN-based approaches have severe overhead of inference: (i) they require special procedures to convert abduction problems into deduction problems because of the deductive nature of MLNs, and (ii) they need grounding for inference.

To emulate abduction in the deductive framework, the pioneering work of MLN-based abduction [Kate and Mooney, 2009] exploits the reverse implication

of the original axioms, and uses the additional axioms to emulate *explaining away* effect (i.e. inferring one cause makes another cause less probable). For example, suppose $B = \{p_1 \rightarrow q, p_2 \rightarrow q,\ p_3 \rightarrow q\}$. Then, B is not used in MLN background knowledge base as it is: B is converted into the following set of logical formulae: $\{q \rightarrow p_1 \vee p_2 \vee p_3,\ q \rightarrow \neg p_1 \vee \neg p_2,\ q \rightarrow \neg p_1 \vee \neg p_3\}$. As the readers can imagine, MLN-based approach suffers from the inefficiency of inference due to the increase of converted axioms. In addition, to the best of our knowledge, most of the existing approaches for maximum-a-posterior (MAP) inference for MLN [Riedel, 2008; Singla and Domingos, 2006, etc.] need (partial) grounding of axioms, which makes inference prohibitively slow.

In terms of the applications, there are also a lot of researches that exploit abduction in many fields. For example, in Systems Biology, abduction is used for discovering scientific knowledge, such as causal relationships from genotype to phenotype, or modeling inhibition in metabolic networks [Doncescu et al., 2008; Tamaddoni-Nezhad et al., 2006, etc.].

3.6 Conclusion

We have proposed an ILP-based formulation for cost-based abduction in first-order predicate logic. Compared to prior work, our method is more expressive and efficient, and its theoretical correctness is guaranteed. Although FOL reasoning is computationally expensive, the proposed optimization strategy using Cutting Plane Inference brings us to a significant boosting of the efficiency of the reasoner. We have evaluated our method on two datasets, including real-life problems (i.e. RTE dataset with axioms generated from WordNet and FrameNet). Our evaluation revealed that our inference method CPI4CBA was highly efficient than other existing systems. The abductive inference engine presented in this chapter is made publicly available.[1]

[1] http://github.com/henry-n700/

Chapter 4

Online Large-margin Weight Learning for Cost-based Abduction

Less attention has been paid to how to automatically learn a function, which rank candidate explanations in order of their plausibility (henceforth, we call it the *score function*). To apply abductive inference to a wide range of tasks, this non-trivial issue needs to be addressed because the criterion of plausibility is highly task-dependent. A notable exception is a series of studies in the context of Statistical Relational Learning [Blythe et al., 2011; Kate and Mooney, 2009; Raghavan and Mooney, 2010; Singla and Domingos, 2011], where they emulate abduction in the probabilistic deductive inference framework, Markov Logic Networks (MLNs) [Richardson and Domingos, 2006], or Bayesian Logic Programs [Kersting and Raedt, 2001]. These approaches can exploit several choices of machine learning methods originally developed for probabilistic models [Huynh and Mooney, 2009; Lowd and Domingos, 2007]. However, emulating abduction in these approaches has severe overhead. For example, the emulation in MLNs requires special procedure to convert abduction problems into deduction problems because MLNs are deductive inference framework in nature. This conversion process generates a large number of axioms, and hence hampers the application of MLN-based approaches to larger problems (see Sec. 4.5 for more detail). Since

inference is a subroutine of learning procedure, learning is also intractable on large dataset, as reported in [Singla and Domingos, 2011].

In this chapter, we propose a supervised learning approach for first-order logic-based abduction, extending the tractable first-order abductive inference engine [Inoue and Inui, 2012]. In order to apply abduction to a wide range of tasks, we support two kinds of gold-standard explanations as training examples: *exactly-specified*, or *partially-specified*. Given *exactly-specified* gold-standard explanations, our framework trains the score function so that it ranks the given explanation itself as the best explanation. Given *partially-specified* gold-standard explanations, on the other hand, the framework trains the score function so that it ranks *any* explanation that *includes* the gold-standard explanation as the best explanation. It is useful to support *partially-specified* gold-standard explanations, because one might want to use abduction for a specific task, where the *subset* of the best explanation is used as the output label of the task. In the case of plan recognition, for example, one might want a system to output *any* explanation that *includes* the correct plan literals, and does not care about any other types of literals in the explanation.

We formulate these learning problems as discriminative structured learning with latent variables. More specifically, we model the score function as a weighted linear feature function, and then apply Passive Aggressive algorithm [Crammer et al., 2006], an online large-margin training algorithm, to tune the weights.

In the rest of this chapter, we first formalize the abductive reasoning problem as a structured prediction with a weighted linear model, and then define the weight learning problem (Sec. 4.1). We show how to use the weighted linear feature function in the ILP-based formulation (Sec. 4.2), and then show how to learn the weights by instantiating Passive Aggressive algorithm. We start with the simple case where exactly-specified gold-standard explanations are given (Sec. 4.3.1), and then describe a learning framework for partially-specified gold-standard explanations (Sec. 4.3.2)

4.1 Problem Formulation

We first generalize the cost function of abudction with a weighted linear model. Henceforth, we use the term *score function*, following the convention of statistical machine learning study. Let $\mathbf{\Phi}(H) = \{\phi_1(H), \phi_2(H), ..., \phi_n(H)\}$ be a n-dimensional feature vector of an explanation H, and $\mathbf{w} = \{w_1, w_2, ..., w_n\}$ be a n-dimensional weight vector. We then define the score function as follows:

$$score(H; \mathbf{w}) = \mathbf{w} \cdot \mathbf{\Phi}(H) = \sum_{i=1}^{n} w_i \cdot \phi_i(H) \tag{4.1}$$

We refer to $\mathbf{w}$ as the *parameter* of score function. We assume each element $\phi_i(H)$ to be the following:

$$\phi_i(H) = \begin{cases} V_i & \text{if } H \cup B \models C_i; \\ 0 & \text{otherwise,} \end{cases} \tag{4.2}$$

where V_i is a real-valued constant, and C_i is a first-order logical formula where each element is a literal or substitution included in H. We call V_i the *feature value*, $\phi_i(H)$ the *feature function*, and C_i the *feature condition*. The feature vector is designed by a user. For example, one might create a feature function ϕ_i such that $(V_i, C_i) = (1, x = y \wedge cat(x) \wedge dog(y))$. The task of abductive reasoning is then formalized as follows:

$$H = \arg\max_{H \in \mathcal{H}_{O,B}} score(H; \mathbf{w}) = \arg\max_{H \in \mathcal{H}_{O,B}} \mathbf{w} \cdot \mathbf{\Phi}(H) \tag{4.3}$$

Notice that this formulation is equivalent to a structured prediction problem (or multi-class classification problem), where the input is O, B, and the set of possible output structures (or classes) is $\mathcal{H}_{O,B}$. We find the best H in the modified ILP-based framework, which is described in the next section.

Let us formalize the supervised learning problem of first-order logic-based abduction. Let $\mathbb{D} = \{(O_i, H_i)\}_{i=1}^{n}$ be a set of training examples, where O_i is an observation (i.e. input) and H_i is either exactly-specified, or partially-specified gold-standard explanation for O_i. Based on the definition in Sec. 3.1, we assume

that O_i and H_i are given by a set of literals or substitutions. The goal of supervised learning is to learn $score(H; \mathbf{w})$, which has minimal prediction errors on $\mathbb{D}$. To achieve this goal, we estimate a weight vector $\mathbf{w}$ that minimizes the value $\sum_{i=1}^{n} \Delta(\hat{H}_i, H_i)$, where $\hat{H}_i$ is the best explanation for O_i inferred by the system, and $\Delta(\hat{H}_i, H_i)$ is a non-negative function that measures the difference between $\hat{H}_i$ and H_i. Henceforth, we call $\Delta(\hat{H}_i, H_i)$ the *loss function*. Because the definition of loss is task-dependent, the loss function is designed by the user. The simple example of loss function for exactly-specified gold-standard explanations is the following (a.k.a 0-1 loss function):

$$\Delta(\hat{H}_i, H_i) = \begin{cases} 1 & \text{if } \hat{H}_i \neq H_i; \\ 0 & \text{otherwise (i.e. } \hat{H}_i = H_i) \end{cases} \tag{4.4}$$

In this paper, we assume that there is enough knowledge to infer the gold-standard explanation for each problem (*the knowledge completeness assumption*). If this assumption were not satisfied, which means that the gold-standard explanation is not included in the candidate explanations, then we could not infer the gold-standard explanation even if we change the weight vector.

4.2 ILP-based Abduction with Weighted Linear Model

In order to exploit the weighted linear feature function as the score function, we replace the ILP-based objective function (3.8) with equation (4.5). We introduce new ILP variables $f_i \in \{0, 1\}$ such that $f_i = 1$ if and only if the feature condition C_i is entailed by $H \cup B$; $f_i = 0$ otherwise. The extended ILP objective function is as follows:

$$\text{max. } score(H; \mathbf{w}) = \sum_{i=1}^{n} w_i \cdot (V_i \cdot f_i) \tag{4.5}$$

Following the definition of f_i above, we associate the feature condition C_i with the assignment of f_i by introducing new ILP constraint so that $f_i = 1 \Leftrightarrow H \cup B \models$

C_i. In general, however, this association cannot be represented as a single ILP constraint. Therefore, we first decompose C_i into a *Conjunctive Normal Form* $\mathrm{CNF}(C_i)$, a set of disjunctive clause, and then introduce ILP constraints for each disjunctive clause.

Let D_i^j be the j-th disjunctive clause in $\mathrm{CNF}(C_i)$. For all $j \in \{1, 2, ..., |\mathrm{CNF}(C_i)|\}$, we first introduce new ILP variable $f_i^j \in \{0, 1\}$ such that $f_i^j = 1 \Leftrightarrow H \cup B \models D_i^j$. To allow to set $f_i^j = 1$ iff $H \cup B \models D_i^j$, we impose the following ILP constraint:

$$0 \leq |l(D_i^j)| f_i^j - [\sum_{L \in l(D_i^j)} I(L)] \leq |l(D_i^j)| - 1, \tag{4.6}$$

where $l(D_i^j)$ is a set of literals or substitutions in D_i^j, and $I(L)$ is a function that returns h_L if L is a literal; s_L if L is a positive substitution; $1 - s_L$ if L is a negative substitution. On the most-right of the term, we add -1 because at least one $L \in l(D_i^j)$ must be hypothesized (remember that D_i^j is a disjunctive clause) when $f_i^j = 1$.

Finally, to ensure that $f_i = 1$ iff $f_i^j = 1$ for all $j \in \{1, 2, ..., |\mathrm{CNF}(C_i)|\}$, we introduce the following ILP constraint:

$$-|\mathrm{CNF}(C_i)| + 1 \leq |\mathrm{CNF}(C_i)| f_i - \sum_{j=1}^{|\mathrm{CNF}(C_i)|} f_i^j \leq 0 \tag{4.7}$$

Note that we are able to use a constant instead of f_i in equation (4.5) when the value of feature is decidable from observations (i.e. $O \models C_i$). In this case, the constraints (4.6), (4.7) need not be introduced.

Let us describe the ILP constraints (4.6), (4.7) with an example. Suppose that we have the feature condition $C_k = \neg p(x) \wedge (p(y) \vee q(y) \vee x \neq y)$ for k-th feature. The CNF of this formula is $\{\neg p(x), p(y) \vee q(y) \vee x \neq y\}$. We thus introduce two ILP variables for each clause: $f_k^1, f_k^2 \in \{0, 1\}$, and then introduce the ILP constraints $f_k^1 - h_{\neg p(x)} = 0$ (i.e. $f_k^1 = 1 \Leftrightarrow H \cup B \models \neg p(x)$), and $0 \leq 3f_k^2 - [h_{p(y)} + h_{q(y)} + (1 - s_{x,y})] \leq 2$ (i.e. $f_k^2 = 1 \Leftrightarrow H \cup B \models [p(y) \vee q(y) \vee x \neq y]$). Finally, we introduce $-1 \leq 2f_k - (f_k^1 + f_k^2) \leq 0$ to ensure that $f_k = 1 \Leftrightarrow f_k^1 = 1 \wedge f_k^2 = 1$.

Algorithm 4 learnExact(training examples $\mathbb{D}$, background knowledge B, int N, double C)

1: $\mathbf{w} \leftarrow \mathbf{0}$
2: **for** $n = 1$ to N **do**
3: **for all** $(O_i, H_i) \in \mathbb{D}$ **do**
4: $\hat{H} \leftarrow \arg\max_{H \in \mathcal{H}_{O_i,B}} score(H; \mathbf{w})$
5: **if** $\hat{H} \neq H_i$ **then**
6: $\tau \leftarrow \min(C, \frac{score(H_i;\mathbf{w}) - score(\hat{H};\mathbf{w}) + \Delta(\hat{H},H_i)}{||\mathbf{\Phi}(\hat{H}) - \mathbf{\Phi}(H_i)||^2})$
7: $\mathbf{w} \leftarrow \mathbf{w} + \tau(\mathbf{\Phi}(H_i) - \mathbf{\Phi}(\hat{H}))$
8: **end if**
9: **end for**
10: **end for**
11: **return** $\mathbf{w}$

4.3 Online Large-margin Weight Learning

4.3.1 Learning from Exactly-specified Explanations

In order to train the weight vector $\mathbf{w}$, we employ Passive-Aggressive (PA) algorithm [Crammer et al., 2006], which is a supervised large-margin online learning algorithm applicable to a wide range of linear classifiers ranging from binary classifiers to structured predictors. The motivation is that (i) an online learning makes our framework scalable, and (ii) it has been empirically shown that large-margin approaches demonstrate a superior generalization ability on unseen datasets. In this section, we consider the simplest setting where *exactly-specified* explanations are given as training examples. The framework learns the score function so that it ranks the given explanation itself as the best explanation.

Algorithm 4 depicts our learning algorithm. Every time we receive a training instance (O_i, H_i) from a set $\mathbb{D}$ of training instances, we first find the highest-score explanation $\hat{H}$ given the current weight vector (line 4). If the current prediction has a prediction error, we train the weight vector (line 5–8). A new weight vector $\mathbf{w}$ should satisfy the following conditions: (i) $score(H_i; \mathbf{w})$ is greater than $score(\hat{H}; \mathbf{w})$ by at least a margin $\Delta(\hat{H}, H_i)$, and (ii) the difference between the current weight vector $\mathbf{w}'$ and the new weight vector $\mathbf{w}$ is minimal. In line 6, we calculate how much $\mathbf{w}$ should be corrected, where C is a parameter of PA

algorithm, meaning the aggressiveness of weight updates. Intuitively, the more different $\hat{H}$ and H_i are, the larger an ensured margin is.

4.3.2 Learning from Partially-specified Explanations

Let us consider the case where we use abduction for a specific task, and the *subset* of the best explanation is used as the output label of the task. In plan recognition, for example, one might use only plan literals (i.e. literals that represent a plan) in the best explanation to decide the system output, and might not care about any other types of literals in the explanation. In this situation, the learning framework is required to have the capability of learning the score function from *partially-specified* gold-standard explanations: training a weight vector that can rank *any* explanation that *includes* the gold-standard explanation as the best explanation, because one wants the system to output *any* explanation that *includes* the correct plan literals. Of course, one could exhaustively give all explanations that include the correct plan literals as exactly-specified explanations, but it is intractable in many cases due to the exponential growth of the number of candidate explanations.

Therefore, in this section, we extend the learning algorithm in the previous section to allow the setting where H_i is partially-specified gold-standard explanation. We formulate the learning problem as a discriminative structured learning with latent variables [Cherry and Quirk, 2008; Felzenszwalb et al., 2010; Yu and Joachims, 2009] etc., where the output label is a set of literals that are specified in H_i, and the rest are regarded as latent variables.

Algorithm 5 depicts the extended learning algorithm. The key extensions from Algorithm 4 are two folds: (i) we update the weight vector if the partially-specified gold-standard explanation H_i is *not included in* the current prediction $\hat{H}$ (line 5–9), and (ii) we perform *latent variable completion*, the inference to complete the unspecified part of the partially-specified gold-standard explanation H_i (line 6). We refer to the completed explanation as the *pseudo exactly-specified explanation*. Note that one can use k-best completed explanations as pseudo exactly-specified explanations, instead of using one completed explanation. In future, we will compare the performance of the k-best explanations approach with the 1-best

Algorithm 5 learnPartial(training examples $\mathbb{D}$, background knowledge B, int N, double C)

Initialize $\mathbf{w}$
for $n = 1$ to N **do**
  for all $(O_i, H_i) \in \mathbb{D}$ **do**
    $\hat{H} \leftarrow \arg\max_{H \in \mathcal{H}_{O_i,B}} score(H; \mathbf{w})$
    if $H_i \not\subseteq \hat{H}$ **then**
      $\overline{H} \leftarrow \arg\max_{H \in \mathcal{H}_{O_i,B}} score(H; \mathbf{w})$ subject to $H_i \subseteq H$
      $\tau \leftarrow \min(C, \frac{score(\overline{H};\mathbf{w}) - score(\hat{H};\mathbf{w}) + \Delta(\overline{H},H_i)}{||\mathbf{\Phi}(\hat{H}) - \mathbf{\Phi}(\overline{H})||^2})$
      $\mathbf{w} \leftarrow \mathbf{w} + \tau(\mathbf{\Phi}(\overline{H}) - \mathbf{\Phi}(\hat{H}))$
    end if
  end for
end for
return $\mathbf{w}$

explanation approach.

Latent variable completion In order to infer $\overline{H}$ in latent variable completion, we follow Yamamoto et al. [2012]'s learning framework for abduction, where $\overline{H}$ is the highest-score explanation among candidate explanations that *are the super set of* H_i. To find such $\overline{H}$, we perform abduction with (O_i, B), satisfying the following two constraints: (i) for all *literal* $L \in H_i$, there exists a literal U and a set of substitutions θ in $\overline{H}$ such that $U\theta = L$ (i.e. $\overline{H} \models H_i$), and (ii) for all *substitution* $x = y \in H_i$, $x = y$ must be hypothesized in $\overline{H}$ i.e. $(\overline{H} \models x = y)$. These constraints ensure that the best explanation for O_i entails H_i. To impose constraint (i), we create the feature function $\mathbf{\Phi}_L(H)$ for all $L \in H_i$, which returns $-\infty$ if none of the literals unifiable with L are hypothesized:

$$\mathbf{\Phi}_L(H) = \begin{cases} -\infty & \text{if } H \not\models \bigvee_{L' \in H}(\text{UNIF}(L, L') \wedge L'); \\ 0 & \text{otherwise,} \end{cases} \tag{4.8}$$

where $\text{UNIF}(L, L')$ is true only if $L \equiv p(x_1, x_2, ..., x_n)$ and $L' \equiv q(y_1, y_2, ..., y_n)$ are unifiable (i.e. $p \equiv q$ and $x_1 = y_1 \wedge x_2 = y_2 \wedge ... \wedge x_n = y_n$); false otherwise. For constraint (ii), we add the following ILP constraints: $s_{x=y} = 1$ for all $x = y \in H_i$, and $s_{x=y} = 0$ for all $x \neq y \in H_i$. Note that the score of $\overline{H}$ will be $-\infty$ when

Algorithm 6 distLearnPartial(training examples $\mathbb{D}$, background knowledge B, int N, int S, int PA_N, double PA_C)

Shard $\mathbb{D}$ into S pieces $\mathbb{D} = \{\mathbb{D}_1, \mathbb{D}_2, ..., \mathbb{D}_S\}$
$\mathbf{w} \leftarrow \mathbf{0}$
for $i = 1$ to N **do**
  for $s = 1$ to S **do**
    $\mathbf{w}^{(i,s)} \leftarrow learnPartial(\mathbb{D}_s, B, PA\_N, PA\_C, \mathbf{w})$
  end for
  $\mathbf{w} = \sum_s \mu_{i,s} \mathbf{w}^{(i,s)}$
end for
return $\mathbf{w}$

knowledge complete assumption is not satisfied. We skip the weight update if the score is $-\infty$.

4.3.3 Distributed Learning

To make the framework more scalable, we implemented the training algorithm in a distributed structure learning framework for perceptrons, following [McDonald et al., 2010]. The algorithm is shown in Algorithm 6. In the distributed learning framework, training dataset is divided into S pieces. For each piece, we independently run the learning procedure in parallel. Finally, we merge the weight vectors learned from each piece. According to McDonald et al., the convergence property of this algorithm is also theoretically guaranteed, when we use Passive Aggressive algorithm [Crammer et al., 2006].

4.4 Evaluation

In this section, we evaluate our online large-margin learning algorithm in two applications to answer the following questions: (i) does the weight vector trained by partially-specified explanations indeed give predictive performance better than the untuned weight vector does? (ii) can machine learning-based abductive reasoning be combined with the powerful existing feature-based classifiers (e.g. Support Vector Machines [Vapnik, 1995a]) for boosting predictive performance? For all experiments, we run our own implementation for the extended version of ILP-

Table 4.1: Feature set used for abductive story understanding.

Feature	Description
PREDICATES_HYPOTHESIZED	a set of predicate names of literals that are hypothesized.
PREDICATES_EXPLAINED	a set of predicate names of literals that are explained by at least one set of literals.
PREDICATES_UNIFIED	a set of predicate names of literals that have at least one equivalent literal in a explanation.
AXIOMS_SATISFIED	a set of names of axioms that are satisfied by a explanation.

based reasoner shown in Sec. 4.2. The implementation is made publicly available on the web.[1] We used a 12-core Opteron 6174 (2.2GHz) 128 GB RAM machine. We used Gurobi optimizer 5.0[2] as an ILP solver, and 8 cores for solving ILP problems in parallel processing. The parameter C of PA algorithm is set to 1.0 in the experiments.

4.4.1 Story Understanding

The task of story understanding is to abductively infer the top-level plans of characters from observed actions. For example, given "*Bill went to the liquor-store. He pointed a gun at the owner*," we need to infer *Bill*'s plan, e.g. *Bill* is robbing at *the liquor store*. By evaluating our algorithm on this task, we want to empirically check whether our algortihm has the capability to learn the signals of "good" explanation from *partially-specified* gold-standard explanations or not.

We used Ng and Mooney [1992]'s story understanding dataset, which is widely used for evaluation of abductive plan recognition systems [Kate and Mooney, 2009; Raghavan and Mooney, 2010; Singla and Domingos, 2011]. The dataset consists of development set and test set, each of which includes 25 pairs of observed actions and its gold-standard plan.[3] In the dataset, the actions and gold-

[1] http://github.com/naoya-i/henry-n700/

[2] http://www.gurobi.com/

[3] To the best of our knowledge, this dataset is a only public dataset that provides a complete test environment for abduction, although it is small. We plan to create the bigger dataset for future evaluation.

standard plans are given by a set of first-order literals (e.g. $\{inst(get2, getting)$, $agent\_get(get2, bob2), name(bob2, bob)\}$). The dataset contains on average 12.6 literals in the actions, and 12.0 literals in the gold-standard plans. The dataset also provides the background knowledge base, which contains 107 first-order logical Horn clauses (e.g. $inst(R, robbing) \wedge get\_weapon\_step(R, G) \rightarrow inst(G, getting)$). We use the development set for training, and the test set for measuring predictive performance. We gave the gold standard plan literals as partially-specified gold-standard explanations for training.

To perform plan recognition, we apply abduction with the background knowledge base, giving the observed actions as observations. We summarize the feature vector used for this setting in Table 4.1. To capture the feature of explanations, we introduce a feature that represents what kinds of literals are included (PREDICATES_HYPOTHESIZED), and explained (PREDICATES_EXPLAINED) in an explanation. We also incorporate the information of axioms satisfied by an explanation (AXIOMS_SATISFIED). PREDICATES_UNIFIED feature captures the following intuition: the information that is supported by many observations (i.e. the situation where the same kind of literal is hypothesized from multiple observations) is more reliable. All the features are encoded by 0-1 features, and each one represents whether each element is included in an explanation.

For the loss function, we want to measure the difference between predicted explanation $\overline{H}$ and the gold-standard explanation H, in terms of plan literals. We used the following function:

$$\Delta(\overline{H}, H) = |H| - |H \cap \overline{H}| + n(\overline{H}), \tag{4.9}$$

where $n(\overline{H})$ is the number of plan literals in $\overline{H}$ that are not included in H. We considered 10 types of literals as plan literals, following Singla and Domingos [2011]. It is clear that this function is a non-negative, and its value is zero iff (i) $H \subseteq \overline{H}$, and (ii) $\overline{H}$ includes plan literals only specified in H.

For evaluating the prediction performance of our system, we focused on how well the system infers plan literals, including their role fillers, following Singla and Domingos [2011]. More specifically, we use precision (ratio of inferred literals that are correct), recall (ratio of correct literals that are inferred by the system),

Table 4.2: Performance of plan recognition in two settings.

	Logical Abduction				Trained			
	Loss	P	R	F	Loss	P	R	F
Closed Test	0.24	0.20	0.40	0.27	**0.12**	**0.35**	**0.69**	**0.46**
Open Test	0.26	0.18	0.44	0.25	**0.18**	**0.28**	**0.57**	**0.37**

and F-measure (harmonic mean of precision and recall), because the gold data often has multiple plan literals.

Results and discussion: To see the effect of weight learning, we show the value of loss function averaged for all the problems, and predictive performances for closed test and open test in Table 4.2. We consider two settings here. In Logical Abduction setting, we try to simulate classical logical abduction that favors the fewer number of elemental explanations: we thus set -1.0 to Predicates_Hypothesized, and 1.0 to Predicates_Explained and Predicates_Unified, and do not tune the weights. In Trained setting, we used our learning procedure for tuning a weight vector.[1] In both tests, Table 4.2 indicates that the training algorithm reduced the loss value than classical logical abduction did, so that it improved the predictive performance. The results of open test also reveal that our learning algorithm shows the generalization ability to unseen data.

4.4.2 NP Coreference Resolution

Noun-phrase (NP) coreference resolution is the task of identifying the group of NPs that refer to the same entity in the world. For example, in the sentence "*Tim shouted at Ed because he was angry.*", we need to identify the group {*he*, *Tim*}. On the other hand, in the sentence "*Tim shouted at Ed because he crashed the car.*", we need to identify the group {*he*, *Ed*}. As the reader can see, coreference resolution requires commonsense reasoning using world knowledge, such as causal relations of events, and synonymous relations of words, etc.

The question here is: what benefits could we receive from the development of machine learning framework for abduction? Our hypothesis is that combining the

[1] A weight vector is initialized with the zero vector.

learning of logical inference with the existing powerful feature-based classifier (e.g. Support Vector Machines [Vapnik, 1995a]) would improve the performance of knowledge-intensive tasks such as coreference resolution. Therefore, we compare the predictive performance of feature-based classifier with a machine learning-based abductive reasoning procedure combined with the existing feature-based classifiers, using coreference resolution as a test bed. To simulate the feature-based classifiers, we created a feature function for each pair of literals that represent NPs, following the feature set proposed by Soon et al. [2001b], which is widely used as the simple baseline model of coreference resolution. Henceforth, we call it SOON system.

To solve coreference problems with abduction using world knowledge, we adopt the idea of Interpretation as Abduction [Hobbs et al., 1993]. The idea is that the interpretation of sentences is an abductive explanation to the logical forms (LFs) of sentences, where substitutions correspond to the identification of coreference relations. We thus perform abduction with world knowledge, giving the LFs of text as an observation. We then extract substitutions from the best explanation for identifying the coreference relations. For combining the abductive reasoning with SOON system, we use the feature set summarized in Table 4.1 and the feature set of SOON system simultaneously in the score function. The resulting system is called SOON+ABDUCTION.

We use the CoNLL-2011 shared task dataset [Pradhan et al., 2011].[1] We used 100 documents of training dataset for training, and 100 documents from development dataset for testing. We convert the dataset into the logical forms, and encode the gold-standard coreference annotations as substitutions. We then give the substitutions as partially-specified gold-standard explanations. We used Boxer semantic parser [Bos, 2008] for the logical form conversion. As a world knowledge, we used WordNet [Fellbaum, 1998] and FrameNet [Ruppenhofer et al., 2010]. We convert the world knowledge to the form of axioms, such as $synsetX(s) \rightarrow dog(s)$, following Ovchinnikova [2012].

For the loss function, we used a pairwise loss function $\Delta_P(\overline{H}, H) = W_O/T_O$, where T_O is the number of pairs of variables in the observation and W_O is the number of substitutions for observed variables (i.e. variables representing NPs)

[1] http://conll.cemantix.org/2011/.

Table 4.3: Performance of NP coreference resolution, provided by feature-based classifier and abductive reasoner combined with feature-based classifier.

Setting	System	Pairwise Loss
Closed Test	Soon	0.40
	Soon+Abduction	**0.29**
Open Test	Soon	0.55
	Soon+Abduction	**0.48**

in H that disagrees with $\overline{H}$. The pairwise loss function is also used for supervised clustering-based coreference resolution [Finley and Joachims, 2005]. Again, it is clear that this function is a non-negative, and its value is zero iff there are no disagreement.

Results and discussion: Table 4.3 shows the values of pairwise loss function in closed test and open test setting. For Soon+Abduction, we initialized the weight vector with the same value as Logical Abduction setting in the story understanding setting, and then trained the weights. In both settings, the loss of Soon+Abduction system is less than Soon system. This indicates that combining the learning of logical inference using the world knowledge with feature-based classifier has a positive impact to the predictive performance of feature-based classifier. In our future work, we will conduct an additional experiment to check the best way to exploit the world knowldge: comparing the results with the performance of feature-based classifier using the world knowledge as a feature.

4.5 Related Work

Probabilistic logical abduction has been studied in the context of Statistical Relational Learning [Blythe et al., 2011; Kate and Mooney, 2009; Raghavan and Mooney, 2010; Singla and Domingos, 2011] etc. They assume to use the standard learning algorithms of probabilistic models (e.g. EM) for learning the score function. However, the inference of probabilistic models for first-order logical inference is computationally expensive, because the inference is performed on a propositional level. Due to the intractability of inference, some work report

that they could not learn weights on large dataset [Blythe et al., 2011; Singla and Domingos, 2011]. Raghavan and Mooney [2010] propose Bayesian Abductive Logic Programs, which constructs a Bayesian Network by using the backward-chaining procedure similar to the ILP-based approach, but they use a task-specific heuristic rule to unify literals to reduce the computational complexity of inference during the construction of the network. Given much larger and dataset in general domain, their framework would not be a scalable solution. Other researchers [Blythe et al., 2011; Kate and Mooney, 2009; Singla and Domingos, 2011] employ Markov Logic Networks (MLNs) [Richardson and Domingos, 2006] to emulate abductive inference. MLNs provide well-studied software packages of inference and learning; however, MLN-based approaches require special procedures to convert abduction problems into deduction problems because of the deductive nature of MLNs. The pioneering work of MLN-based abduction [Kate and Mooney, 2009] converts background axioms into MLN logical formulae by (i) reversing implication and (ii) constructing axioms representing mutual exclusiveness of explanation (e.g. the set of background knowledge axioms $\{p_1 \rightarrow q, p_2 \rightarrow q, p_3 \rightarrow q\}$ is converted into the following MLN formulae: $q \rightarrow p_1 \vee p_2 \vee p_3$, $q \rightarrow \neg p_1 \vee \neg p_2$, $q \rightarrow \neg p_1 \vee \neg p_3$ etc.). As the readers can imagine, MLN-based approach suffers from the inefficiency of inference due to the increase of converted axioms. In addition, the current solution of MAP inference for MLNs, which is needed for the best explanation finding, works on a propositional level. Therefore, learning would not scale to larger problems due to the severe overhead [Inoue and Inui, 2012]. Singla and Domingos [2011] report that their MLN-based abduction models cannot be trained on larger dataset.

As mentioned in Sec. 4.3.2, Yamamoto et al. formulate the learning problem of first-order logic abduction as the framework similar to us. The key difference is that they use score function that is non-linear in terms of weights, and thus use a different optimization strategy for optimizing the weights. Comparing the performance of our work with them is interesting and important future direction. Our work is also related to a structured learning approaches that exploit latent variables, which demonstrate a superior performance in many tasks ranging from natural language processing to graphical processing. For example, Latent Support Vector Machines, a variant of structured learning model with latent variables, is

widely used [Cherry and Quirk, 2008; Felzenszwalb et al., 2010; Yu and Joachims, 2009] etc. for many classification tasks, and shown to outperform the existing systems.

4.6 Conclusion

In this chapter, we have proposed a supervised approach for learning the score function of abduction. We formulated the learning procedure in the framework of structured learning with latent variables. Our approach enables us to learn the score function from partially-specified gold-standards, which is a useful feature in real-life tasks. In our evaluation, we found that our learning procedure can reduce the loss, and improve predictive performance of story understanding tasks in both open test and closed test. We also explored the potential use of machine learning-based abductive reasoning, i.e. the integration of learning of logical inference and feature-based classifiers. The experiments showed that the integration of these two approaches is promising.

Chapter 5

Resolving Direct and Indirect Anaphora with Feature-based Approach

Anaphora is a phenomenon that a linguistic expression refers to the other linguistic expression. A referring expression is called an *anaphor*, and its referent is called an *antecedent*. In nominal anaphora, an anaphor and its antecedent in the preceding discourse hold either a *direct* anaphoric relation or an *indirect* relation. *Direct anaphoric relation* refers to a link in which an anaphor and an antecedent are in such a relation as *synonymy* and *hypernymy/hyponymy*, as in *house–building*. *Indirect anaphoric relation*, on the other hand, refers to a link in which an anaphor and an antecedent have such relations as *meronymy/holonymy* and *attribute/value* as in *ticket–price*. For the other case, a noun phrase occasionally holds an *exophoric* relation to an antecedent that lies outside the discourse that the noun phrase presents. The process of identifying such anaphoric relation is called *anaphora resolution*.

In this chapter, we conduct the case study on *anaphora resolution*, in order to give a detailed comparison of feature-based and inference-based approaches from both the qualitative and empirical perspective. We propose a feature-based anaphora resolution model in this chapter, and then discuss the problems of feature-based approaches. To improve the performance, we propose an alternative

model of anaphora resolution, which is an inference-based approach in the next chapter.

5.1 Preliminary

Anaphora resolution has been studied intensively in recent years because of its significance in many natural language processing (NLP) applications such as information extraction and machine translation. Recent studies in anaphora resolution have proposed the resolution frameworks for both of direct (e.g., coreference, pronoun resolution) and indirect anaphoric cases (e.g., bridging reference [Clark, 1977] resolution), placing the main focus on the *direct* anaphoric case [Iida et al., 2005; Poesio et al., 2004; Soon et al., 2001a, etc.]. The identification of exophoric relations, in contrast, has been paid little attention in the literature. Anaphoricity determination, which is the task of determining whether an anaphor has an antecedent in the preceding discourse or not, is related to identifying exophoric relations, but the methods for anaphoricity determination are not designed to explicitly capture exophoric relations because they are tuned for finding noun phrase coreference chains in discourse.

However, for the practical use of anaphora resolution, we need to solve the following non-trivial problem: in a real text, anaphors such as noun phrases can occur as either direct anaphoric, indirect anaphoric or exophoric relations, which is not easy to disambiguate from its surface expression. That is, in anaphora resolution, it is necessary to judge what kind of anaphoric relation is used to tie an anaphor and its (potential) antecedent (henceforth, we call this task *anaphora type classification*). In fact, our corpus analysis (detailed in Section 5.4) shows that more than 50% of noun phrases modified by a definiteness modifier (we call such noun phrases *definite noun phrase*) have non-trivial ambiguity in terms of the anaphora types that have to be classified for each given text. Given these issues, we decompose the task of nominal anaphora resolution as a combination of two distinct but arguably interdependent subtasks.

- *Antecedent selection*: the task of identifying the antecedent of a given anaphor, and

- *Anaphora type classification*: the task of judging what kind of *anaphora type* is used for a given anaphor, i.e., classifying a given anaphor into *direct anaphoric*, *indirect anaphoric* or *exophoric*.

Given this task decomposition, three unexplored issues immediately come up:

Issue 1. Whether the model for antecedent selection should be designed and trained separately for direct anaphora and indirect anaphora or whether it can be trained as a single common model;

Issue 2. What contextual information is useful for determining each of the anaphora types;

Issue 3. How the two subtasks can be best combined (e.g., which subtask should be carried out first).

In this chapter, we explore these issues taking Japanese as our target language. Specifically, we focus on anaphora resolution for noun phrases modified by a definiteness modifier, as detailed in the next section.

5.1.1 Definition of Anaphora Type

As mentioned, an anaphor can hold a direct or indirect relation with its antecedent. Occasionally, an anaphor refers to an antecedent that is not in the same discourse. The terms *direct anaphora* and *indirect anaphora* have been used to denote some different anaphoric phenomena in previous work, e.g. *direct anaphora* in [Vieira and Poesio, 2000] indicates only the reference that an anaphor and its antecedent have identical head words, whereas *direct anaphora* in [Mitkov et al., 2000] includes a synonymous or generalization/specialization link of an anaphor and its antecedent. As a result, we redefine the following three *anaphora types* to denote the use of anaphoric expressions in our classification task:

- *direct anaphora*: An anaphor refers to its antecedent directly. In example (1), "そのCD" (*the CD*) refers to "彼女の新しいアルバム" (*her new album*) directly.

(1) 彼女の新しいアルバム$_{(i')}$が昨日発売された．早くその CD$_{(i)}$が欲しい．
Her new album$_{(i')}$ was released yesterday. I want to get the CD$_{(i)}$ as soon as possible.

- *indirect anaphora*: An anaphor has an antecedent related with the anaphor rather than referred to, as in example (2).

(2) そのアーティストは新曲$_{(i')}$を発表した．早くそのCD$_{(i)}$が欲しい．
The artist announced her new song$_{(i')}$. I want to get the CD$_{(i)}$ as soon as possible.

"その CD" (*the CD*) refers to *her new song* indirectly. The discourse entity that directly corresponds to "その CD" (*the CD*) is not in the preceding sentence; instead 新曲 (*her new song*) is considered as an antecedent of "その CD" (*the CD*) because it is associated with "その CD" (*the CD*).

- *exophora*: An anaphor that has no antecedent in a text is regarded as exophoric. An exophoric expression is typically used in newspaper articles; for instance, "その日" (*the day*) refers to the date of the post.

As seen from the above examples (1), (2) and reported in Section 1, the anaphora type can be different for an identical expression. In other words, the anaphora type must be disambiguated taking its appearing context into account.

5.1.2 Definiteness of Japanese Noun Phrase

Definite noun phrase is a noun phrase that describes a specific and identifiable entity in a certain context, as "大統領" (*the president*) referring to "韓国の大統領" (*Korean President*) in example (3):

(3) 今月 4 日，韓国の大統領$_{(i')}$が来日した．大統領$_{(i)}$は翌日の記者会見で，新プランの詳細を語った．
Korean President$_{(i')}$ visited Japan on the 4th this month. The president$_{(i)}$ talked about the details of his new plan at the news conference next day.

On the other hand, *indefinite noun phrase* is a noun phrase that describes a general entity, as "本" (*a book*) in example (4):

(4) コロンビア図書館で本を借りた.
I borrowed a book from Columbia Libraries.

In anaphora resolution, we must determine whether a target noun phrase is definite or indefinite, because indefinite noun phrases have no referent in the text. However, as seen from the above two examples (3) and (4), English noun phrases are easy to determine its definiteness according to the existence of a *definiteness modifier* (*the*, *this*, *that*) [1], whereas Japanese noun phrases are not. For this reason, it is sometimes difficult even for human annotators to determine the definiteness of a bare noun phrase. In this thesis, as the first step toward complete understanding of Japanese nominal anaphora, we focus on anaphora resolution for noun phrases marked with 指示連体詞 (*Kono, Sono, Ano*); "この+NP" (*this NP*), "その+NP" (*the NP*) and "あの+NP" (*that NP*), which account for a large proportion of occurrences of nominal anaphora in Japanese texts.

5.2 Related Work

In this section, we review previous research on anaphora resolution for antecedent selection and anaphora type classification respectively. In Section 5.2.1, we look over how the previous work had taken the approaches to antecedent selection for direct anaphora and indirect anaphora. In Section 5.2.2, we discuss Vieira's work and Nakaiwa's work on anaphora type classification.

5.2.1 Antecedent Selection

A wide range of approaches to antecedent selection has been proposed in earlier work. Note that these studies focus on one side of direct or indirect anaphora, in other words, they are based on the assumption that the system knows that the given anaphor is direct anaphora or indirect anaphora. This motivates us to explore the design of the antecedent selection model (*issue 1*).

[1] In some cases, a noun phrase without a definiteness modifier such as a proper noun, appositional noun phrase can be regarded as definite.

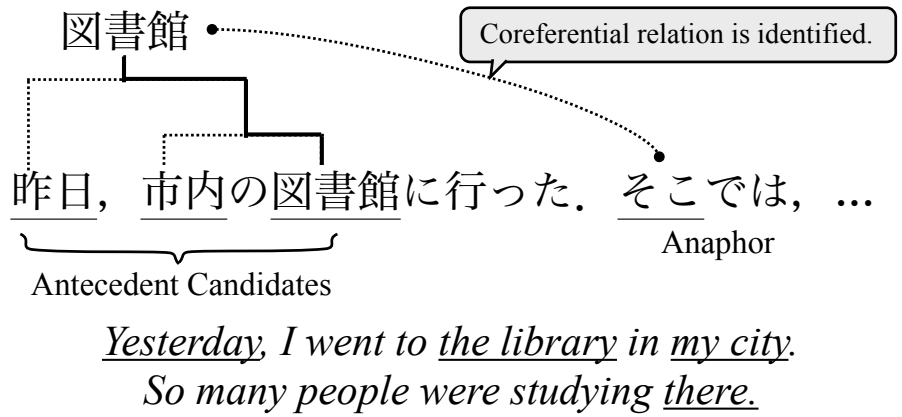


Figure 5.1: Identifying a co-referential relation by the tournament model.

5.2.1.1 Direct Anaphora

There exist two main approaches: rule-based approaches and machine learning-based approaches. In contrast to the rule-based approaches such as Baldwin [1995]; Brennan et al. [1987]; Mitkov [1997]; Okumura and Tamura [1996]; Shalom and J. [1994], empirical, or machine learning-based approaches have shown to be a cost-efficient solution achieving performance that is comparable to the best performing rule-based systems [Ge et al., 1998; Iida et al., 2005; Mccarthy and Lehnert, 1995; Ng and Cardie, 2002, 2001; Soon et al., 2001a; Strube and Muller, 2003; Yang et al., 2003, etc.]. Most of these studies focus only on the coreference resolution task, particularly in the context of evaluation-oriented research programs such as Message Understanding Conference (MUC)[1] and Automatic Content Extraction (ACE)[2].

The state-of-the-art method of Japanese coreference resolution is *tournament model* proposed in [Iida et al., 2005]. The tournament model selects the best candidate antecedent by conducting one-on-one games in a step-ladder tournament. More specifically, the model conducts a tournament consisting of a series of games in which candidate antecedents compete with each other and selects the winner of the tournament as the best candidate antecedent. In order to describe how the

[1]http://www-nlpir.nist.gov/related_projects/muc/index.html
[2]http://www.nist.gov/speech/tests/ace/

tournament model works, suppose we have the following example sentence:

(5) 昨日，市内の図書館$_{(i')}$ に行った．そこ$_{(i)}$ では，たくさんの人が勉強していた．
Yesterday, I went to the library$_{(i')}$ in my city. So many people were studying there$_{(i)}$.

“図書館” (*the library*) and “そこ” (*there*) hold a co-referential relation in example (5). When we apply the tournament model to the anaphor “そこ” (*there*) in example (5), the noun phrases preceding “そこ” (*there*) are regarded as the antecedent candidates, and the tournament model identifies its antecedent “図書館” (*the library*) through the step-ladder tournament as shown in Figure 5.1. In the training procedure of the tournament model, we give instances, each created from an antecedent paired with one other competing candidate. The advantage of this model is that the model can use the information of the two competing candidates at the same time in training and classification, compared to a binary classification approach [Ng and Cardie, 2002; Soon et al., 2001a, etc.]. We adopt the tournament model for creating antecedent selection model, as mentioned in Section 5.3.1

5.2.1.2 Indirect Anaphora

To the contrary, the methods for indirect anaphora resolution have been relatively unexplored compared with direct anaphora. Those works are implemented by rule-based approaches [Bunescu, 2003; Murata et al., 1999; Poesio et al., 1997, etc.] and learning-based approaches [Poesio et al., 2004], encoding the centering theory [Grosz et al., 1995], lexical resources such as WordNet [Fellbaum, 1998] and web-based knowledge. In comparison to direct anaphora, the resolution of indirect anaphora is still a much more difficult task because it is required to capture the wide variety of semantic relations (e.g. *store–the discount*, *drilling–the activity*). For example, Poesio et al. [2002] proposed acquiring the lexical knowledge of the meronymy relations for resolving bridging references [Clark, 1977] by using syntactic patterns such as *the NP of NP* and *NP's NP*.

5.2.2 Anaphora type classification

As mentioned in Section 1, there has been little attention paid to the issue of anaphora type classification. Exceptions can be seen in [Vieira and Poesio, 2000] and [Nakaiwa et al., 1995], and we describe their work in this section. Note that their system carries out anaphora type classification before antecedent selection. However, it remains unexplored how to integrate antecedent identification and anaphora type classification into anaphora resolution, which is to be investigated as *issue 2*.

5.2.2.1 English Definite Description Processing System

Vieira's work (2000) is motivated by corpus study for the use of definite descriptions[1]. Their system does not only find an antecedent but classifies a given definite description into the following three categories.

- *direct anaphora*: subsequent-mention definite descriptions that refer to an antecedent with the same head noun as the description;
- *bridging descriptions*: definite descriptions that either (i) have an antecedent denoting the same discourse entity, but using a different head noun (as in *house ... building*), or (ii) are related by a relation other than identity to an entity already introduced in the discourse;
- *discourse-new*: first-mention definite descriptions that denote objects not related by shared associative knowledge to entities already introduced in the discourse.

Compared with our taxonomy, their definition of *direct anaphora* is restricted to the case where an anaphor and its antecedent have an identical head. Therefore, the other cases (e.g. a pair of *new album* and *the CD*) are not regarded as direct anaphora but such cases are classified into bridging descriptions. The definition of *discourse-new*, on the other hand, refers to the same notion as our definition of *exophora* except that the generic use of the definite article *the* as in *play the*

[1] *Definite description* is a noun phrase marked with *the*.

piano is classified into *discourse-new*. Note that Japanese definiteness modifiers are not used in such a way.

In their work, the system chooses the correct anaphora type of a given definite NP and if possible, finds its antecedent following a set of hand-coded rules on the basis of the lexical and syntactic features. The process can be regarded as four notable steps.

1. The system applies some heuristics exploiting lexical and syntactic features based on [Hawkins, 1978] to detect non-anaphoric cases ('unfamiliar use' or 'larger situation use' in Hawkin's work) to an anaphor. If the test succeeds, it interprets the anaphor as discourse-new.

2. The system tries to find a same-head antecedent (i.e., an antecedent as direct anaphora) from a set of potential candidates appearing in the preceding discourse. If a suitable candidate is found, the system classifies an anaphor as *direct anaphora* and returns the candidate as its antecedent.

3. The rules to recognize discourse-new, such as 'pre-modifier use' and 'proper noun use' (e.g. *the United States*), are applied to an anaphor. If the test succeeds, the anaphor is classified as discourse-new.

4. The system tries to find an NP associated with an anaphor (which is called *an anchor* in their work) in the preceding discourse. If such an NP is found, the anaphor is classified as bridging description and judges the NP as its anchor. Otherwise, the system does not output anymore.

The heuristics to detect non-anaphoric or discourse-new anaphors are based on the syntactic and lexical features, while the rules for direct anaphora and bridging descriptions simply try to find an antecedent. Consequently, their work can be said to focus on detecting discourse-new descriptions compared to our work. They reported their system achieved 57% recall and 70% precision in their empirical evaluation.

5.2.2.2 Extra-sentential Resolution of Japanese Zero Pronouns

Zero pronoun is an invisible pronoun arising from omitting a linguistic expression. In example (6), we can observe a zero pronoun ϕ_i referring to an omitted

expression 私 (*I*).

(6) (ϕ_i は) 近所の本屋で本を買った.
(I) bought a book at a nearby bookstore.

This ellipsis causes a serious problem to NLP applications such as information extraction systems and machine translation systems, and frequently occurs in Japanese. Nakaiwa's work focuses on identifying the semantic type [1] of the referent of such Japanese zero pronouns for machine translation, especially zero pronouns which have no referent inside the discourse. The relatedness to our work is that their work detects an anaphor which has a extra-sentential reference, i.e., classifying anaphor into exophoric or not. Their system determines an anaphor is exophoric when its referent is not found in the discourse, and if it's determined, then identifies its type of referent by using the semantic constraints such as modal expressions, verbal semantic attributes. They reported the accuracy of 85.5% for identifying the type of referent of 196 zero pronouns referring to five types of entity; *I, we, you, it* and a specific person.

As to anaphora type classification, note that (i) their system classifies an anaphor into exophoric or not, and (ii) the clue of its classification is just whether the antecedent is found in the discourse or not.

5.3 Feature-based Anaphora Resolution Models

The purpose of our work is to investigate the three unexplored issues shown in Section 1. First of all, we explain our learning-based antecedent selection models and anaphora type classification models.

5.3.1 Antecedent Selection Model

One issue to explore in antecedent selection is whether a single common model suffices for both direct and indirect anaphora or a separate model should be built for each. In this section, in order to explore *issue 1*, we first design two

[1] I, we, you, it, or a specific person.

different strategies for selecting antecedents in Section 5.3.1.1, and elaborate the antecedent selection models in the rest of section.

5.3.1.1 Mix Strategy and Separate Strategy

From the point of view in which we consider both anaphora types in parallel in an antecedent identification, we can consider the following two strategies as summarized in Figures 5.2 and 5.3.

- *Mix Strategy*: Designing a single model for the resolution of both direct and indirect anaphora. The information to capture a direct-anaphoric antecedent and indirect-anaphoric antecedent is jointly incorporated into a single common model. The model is trained with labeled examples of both direct and indirect anaphora. We call this model as the *mix antecedent selection model*.

- *Separate Strategy*: Preparing a distinct model for each anaphora type separately; i.e., the selection model for direct anaphora and the model for indirect anaphora. Unlike the *mix strategy*, each model incorporates the information to capture an antecedent for each anaphora type separately. In the *direct antecedent selection model*, only the information that captures a direct-anaphoric antecedent is used. In the *indirect antecedent selection model*, on the other hand, only the information for the indirect-anaphoric antecedent is used. For the training, labeled examples of direct anaphora are only used in the *direct antecedent selection model* and labeled examples of indirect anaphora are only used in the *indirect antecedent selection model*.

The separate strategy is expected to be advantageous because useful information for detecting direct-anaphoric antecedents is different from one for indirect-anaphoric antecedents. For example, synonymous relations between anaphor and antecedent are important for selecting direct-anaphoric antecedents. In example (1), an antecedent selection model must know that *CD* and *album* are synonymous. For indirect anaphora, on the other hand, it is required to recognize such semantic relations as *part-whole* and *attribute-value* as shown in example (2), where it is essential that *CD* is semantically related with *song*.

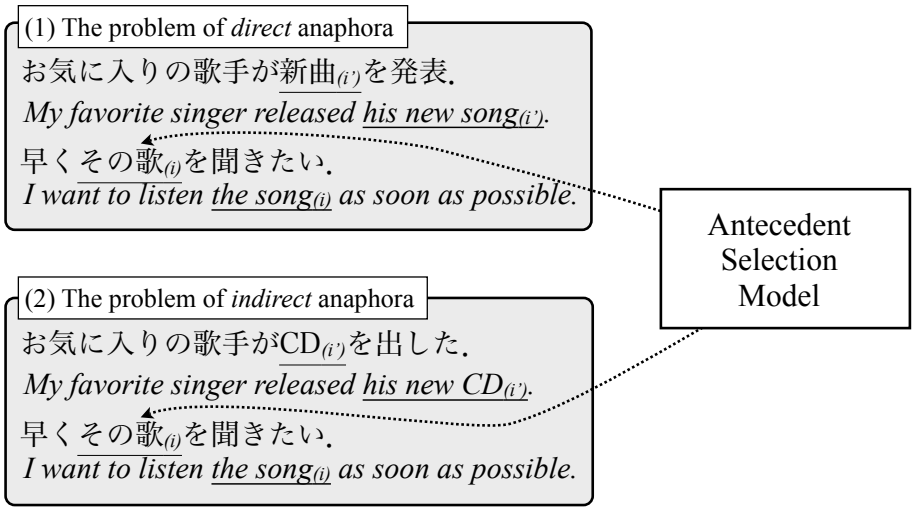


Figure 5.2: Mix strategy for antecedent selection.

The single common model is used for resolving both direct and indirect anaphora.

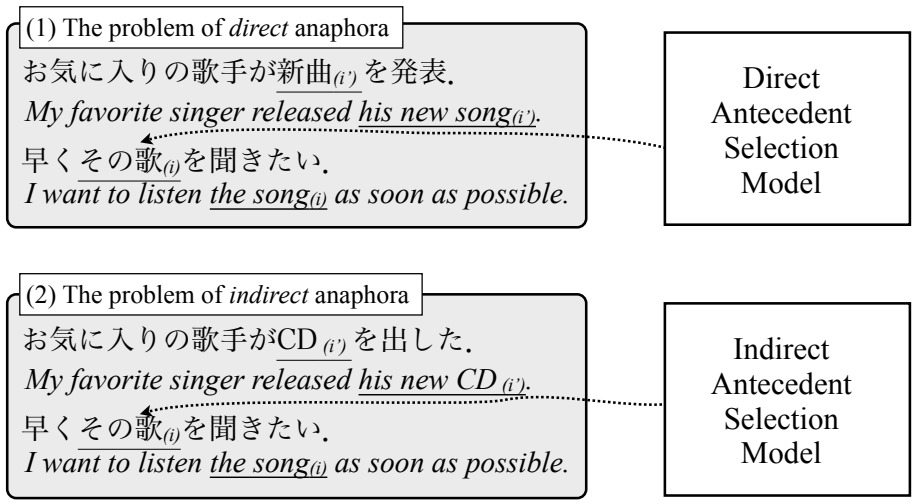


Figure 5.3: Separate strategy for antecedent selection.

The distinct two models are used for resolving each direct and indirect anaphora.

There are a variety of existing machine learning-based methods designed for coreference resolution ranging from classification-based models [Soon et al., 2001a, etc.] and preference-based models [Ng and Cardie, 2001, etc.] to comparison-based models [Iida et al., 2005; Yang et al., 2003, etc.]. Among them, we adopt a state-of-the-art model for coreference resolution in Japanese [Iida et al., 2005], called the *tournament model* because it achieved the best performance for coreference resolution in Japanese as mentioned in Section 5.2.1.1.

5.3.1.2 Training Procedure

Our antecedent selection model learns the preference of the antecedent by the anaphor over the other competing candidate antecedents. Thus, we extract the training instances from 3 elements; an anaphor, its antecedent and the competing candidate. Suppose a text that consists of noun phrases NP_1, NP_2, NP_3, NP_4 and ANP, and let an anaphor ANP and an antecedent NP_2 hold anaphoric relation[1], as shown in Figure 5.4. In this situation, we learn the preference of NP_2 by ANP over the other candidates NP_1, NP_3 and NP_4, so we first extract the training instance $\langle class = right, NP_1, NP_2, ANP \rangle$. The class label denotes which candidate is preferred. In the same manner, we extract the training instances $\langle class = left, NP_2, NP_3, ANP \rangle$ and $\langle class = left, NP_2, NP_4, ANP \rangle$. Figure 5.4 (a) illustrates this procedure.

5.3.1.3 Selection Method

Given an anaphor, the antecedent selection model determines the most likely antecedent by comparing which candidate antecedent is preferred most by a given anaphor in all the candidate antecedents. Our model realizes this decision by conducting a tournament consisting of a series of games in which candidate antecedents compete with each other, taking candidate antecedents in the right-to-left order. Finally, the model identifies the winner of the tournament as the antecedent of the given anaphor. Suppose we have the same text as a text described in Section 5.3.1.2 as shown in Figure 5.4, and we want to select the antecedent

[1] We enumerated only noun phrases as the potential antecedents for convenience. In our evaluations, we include verbal predicates in the list of potential antecedents for such cases as *...we calculate the value in advance. – The precomputation ....*

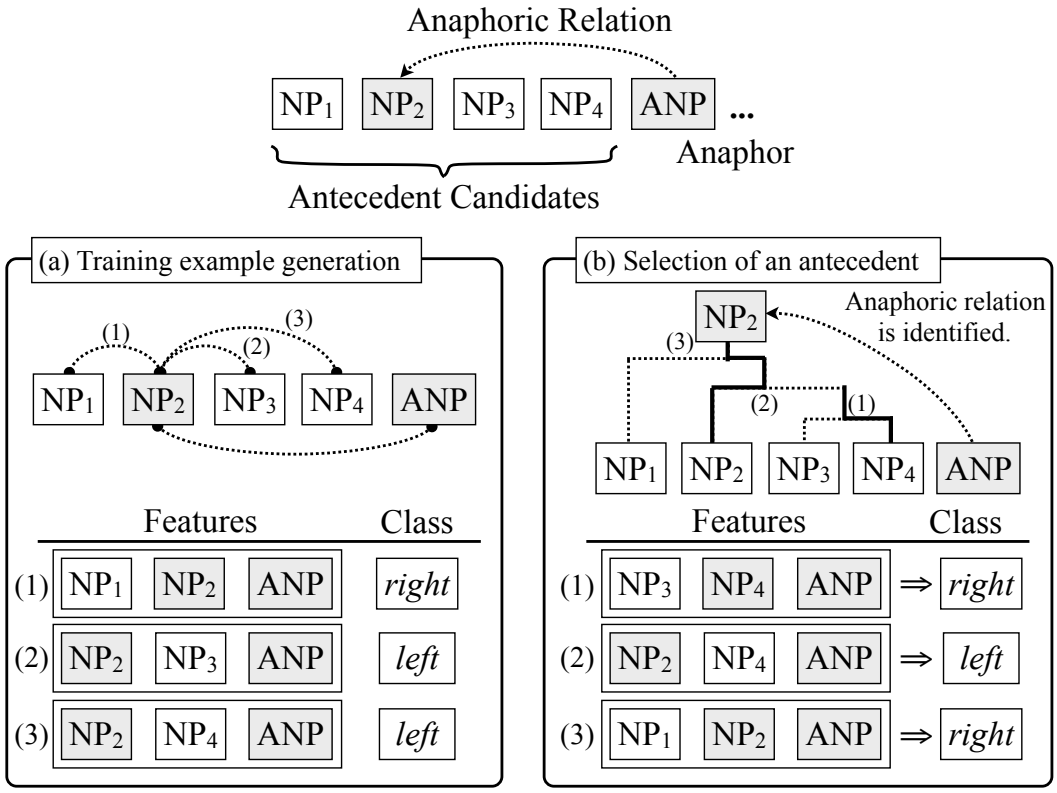


Figure 5.4: The procedure of training example generation and selection for antecedent selection model.

of the anaphor ANP. In this situation, the first game is NP_3 *v.s.* NP_4, so we check whether ANP prefers NP_3 to NP_4, or NP_4 to NP_3 by a binary classifier trained in the manner of Section 5.3.1.2. Thus, we give $\langle NP_3, NP_4, ANP \rangle$ to the binary classifier. Suppose it returned $class = right$; NP_4 won in this game. After this, an winner is compared with the next candidate one by one. That is, we give $\langle NP_2, NP_4, ANP \rangle$ to the classifier and $class = left$ returns. Then we give $\langle NP_1, NP_2, ANP \rangle$ to the classifier and $class = right$ returns. As a result, NP_2 is identified as the antecedent of ANP. This example game is illustrated in Figure 5.4 (b).

5.3.1.4 Feature Set

The feature set for antecedent selection is designed based on the literature of coreference resolution [Denis and Baldridge, 2008; Iida et al., 2005; Ng and Cardie, 2001; Soon et al., 2001a; Yang et al., 2003, etc.] as shown in Table 5.1 and 5.2. In addition, we introduce the following lexical semantic features:

- WN_SEMANTIC_RELATION: In order to capture various semantic relations between an anaphor and its antecedent, we incorporate the binary features that represent the semantic relation found in the Japanese WordNet 0.9 [Isahara et al., 2008][1].

- SYNONYMOUS, IS_HYPONYM_OF_ANAPHOR: We recognize synonymous and hyper-hyponym relations by using a very large amount of synonym and hypernym-hyponym relations (about three million hypernymy relations and two hundred thousand synonymy relations) automatically created from Web texts and Wikipedia [Sumida et al., 2008].

- BGH_ID, BGH_COMMON_ANC: We incorporate the lexical information obtained from the *Bunrui Goi Hyo* thesaurus [NLRI, 1964]. We encode the information as two types: (i) binary features that represent the semantic class ID, and (ii) a real-valued feature that indicates the depth of the lowest common ancestor of an anaphor and its candidate.

[1] http://nlpwww.nict.go.jp/wn-ja/

Table 5.1: Feature set for antecedent selection and the MLAC models.

Feature	Description
DEFINITIVE	1 if C_p is definite noun phrase; else 0.
DEPEND_CLASS*	POS $\in \{NOUN, PREDICATE\}$* of word which C_p depends.
DEPENDED_CLASS*	POS $\in \{NOUN, PREDICATE\}$* of word depending C_p.
ANAPHOR_DM_TYPE	Type of definiteness modifier of ANP.
ANAPHOR_HEAD	Head morpheme of ANP.
ANAPHOR_POS	POS of ANP.
ANAPHOR_CASE	Case particle of ANP.
CANDIDATE_HEAD	Head morpheme of C_p.
CANDIDATE_POS	POS of C_p.
CANDIDATE_NE	Proper noun-type of C_p.
CANDIDATE_CASE	Case particle of C_p.
CANDIDATE_BGH_ID*	The semantic class ID of C_p at the level of a middle grain size defined in Bunrui Goi Hyo.

ANP denotes an anaphor. $C_{p\in\{L,R\}}$ denotes either of the two compared candidate antecedents (C_L and C_R denote the left and right candidate, respectively). '*' denotes features used only in the direct antecedent selection model (ASM), the mix-ASM, the d-MLAC model, or the p-MLAC model. '**' denotes features used only in the indirect-ASM, the mix-ASM, the i-MLAC model, or the p-MLAC model. In the p-MLAC model, the feature set extracted from *direct*-ASM is distinguished from the one extracted from *indirect*-ASM.

Table 5.2: Feature set for antecedent selection and the MLAC models.

Feature	Description
WN_SEMANTIC_RELATION	The semantic relation between ANP and C_p found in WordNet.
STRING_MATCH_TYPE*	The string match type $\in \{HEAD, PART, COMPLETE\}$ if the string of C_p matches the string of ANP; else empty.
SENTENCE_DISTANCE	The number of sentences intervening between C_p and ANP
SIMILARITY*	Distributional similarity between ANP and C_p
PMI**	Point-wise mutual information between ANP and C_p
BGH_COMMON_ANC*	The depth of lowest common ancestor of C_p and ANP in BGH
SYNONYMOUS	1 if C_p and ANP are synonymous; else 0.
IS_HYPONYM_OF_ANAPHOR	1 if C_p is a hyponym of ANP; else 0.
DEPEND_RELATION	Function word when C_L depends on C_R if C_L depends on C_R; else empty.
SENTENCE_DISTANCE	The number of sentences intervening between C_L and C_R
DEPENDED_COUNT_DIFF*	Difference between the count of bunsetsus depending C_L and C_R.

The definition of ANP, $C_{p\in\{L,R\}}$ '*', '**' follows Table 5.1.

- SIMILARITY: To robustly estimate semantic similarities between an anaphor and its candidate antecedent, we adopt the cosine similarity between an anaphor and candidate antecedent, which is calculated from a cooccurrence matrix of $(n, \langle c, v \rangle)$, where n is a noun phrase appearing in an argument position of a verb v marked by a case particle c. The cooccurrences are counted from two decades worth of news paper articles, and their distribution $P(n, \langle c, v \rangle)$ is estimated by pLSI [Hofmann, 1999] with 1,000 hidden topic classes to overcome the data sparseness problem.

- PMI: The degree of indirect-anaphoric association between an anaphor *ANP* and candidate *CND* is calculated differently depending on whether *CND* is a noun or predicate. For the case of a noun, we follow the literature of indirect anaphora resolution [Murata et al., 1999; Poesio et al., 2004, etc.] to capture such semantic relations as *part-whole*. The associativeness is calculated from the cooccurrences of *ANP* and *CND* in the pattern of "*CND の ANP* (*ANP of CND*)". Frequencies of cooccurrence counts are obtained from the Web Japanese N-gram Version 1 [Kudo and Kazawa, 2007]. For the case of a predicate, on the other hand, the associativeness is calculated from the cooccurrences of *ANP* and *CND* in the pattern where *CND* syntactically depends on (i.e. modifies) *ANP* (in English, the pattern like "*ANP that (subj) CND*"). If we find many occurrences of, for example, "闘う (*to fight*)" modifying "夢 (*a dream*)" in a corpus, then "夢 (*a dream*)" is likely to refer to an event referred to by "闘う (*to fight*)" as in (7).

  (7) チャンピオンと闘い$_{(i')}$たい。その夢$_{(i)}$は実現すると信じている。
  I want to fight$_{(i')}$ the champion. I believe the dream$_{(i)}$ will come true.

5.3.2 Anaphora Type Classification Model

As mentioned in Section 5.2.2, the clue of anaphora type classification is the information of a context that precedes an anaphor. However, it is not explored well that which information in the context is useful for anaphora type classification (*issue 2*) in the previous studies. In this section, we consider four machine learning-based models for anaphora type classification in order to find the answer

of *issue 2*. The difference between the contextual clues that each classifier uses is summarized in Table 5.3.

5.3.2.1 No-context Model

This anaphora type classifier determines whether an anaphor bears either direct anaphora, indirect anaphora or exophora, by using only the properties of an anaphor. For training, we give only an anaphor and its annotated anaphora type to the classifier. We illustrate an example of training example generation in Figure 5.5 (a). In Figure 5.5 (a), anaphors ANP_1, ANP_2 and ANP_3 are given to the classifier as its training examples. The feature set used in this model is detailed in Table 5.4.

By comparing this model with the other models, we can see the effect of using contextual information in anaphora type classification.

5.3.2.2 Broad Context Model

This anaphora type classifier determines an anaphora type by using the properties of an anaphor and the lexical and syntactic information from all potential antecedents. For training, we give an anaphor and all the potential antecedents with annotated anaphora type to the classifier. Figure 5.5 (b) illustrates the procedure of the training example generation. In Figure 5.5 (b), for an anaphor ANP_1, anaphor itself ANP_1 and its potential antecedent NP_1, NP_2, NP_3, NP_4 are given to the classifier as its training examples. Our features for learning and classification are summarized in Table 5.4. We use such features as HAS_SYNONYM_OF_ANAPHOR and HAS_STRING_MATCHED, which capture contextual information encoded from all potential antecedents, based on the literature [Vieira and Poesio, 2000, etc.].

5.3.2.3 Most Likely Antecedent Context Model

The Broad Context model described above utilizes all the antecedent candidates as contextual information. Contrary to the Broad Context model, we introduce Most Likely Antecedent Context model which uses only the most likely antecedent(s) as contextual information, instead of all the potential antecedents.

Table 5.3: Summary of the information used in each anaphora type classifier.

Contextual Information	NCM	BCM	MLACM			
			Mix	Direct	Indirect	Parallel
Anaphor	√	√	√	√	√	√
All potential antecedents		√				
Antecedent selected by mASM			√			
Antecedent selected by dASM				√		√
Antecedent selected by iASM					√	√

The mASM, dASM and iASM denote mix, direct and indirect antecedent selection model respectively. The NCM, BCM and MLACM denote No-context, Broad Context and Most Likely Antecedent Context anaphora type classification model described in Section 5.3.2.

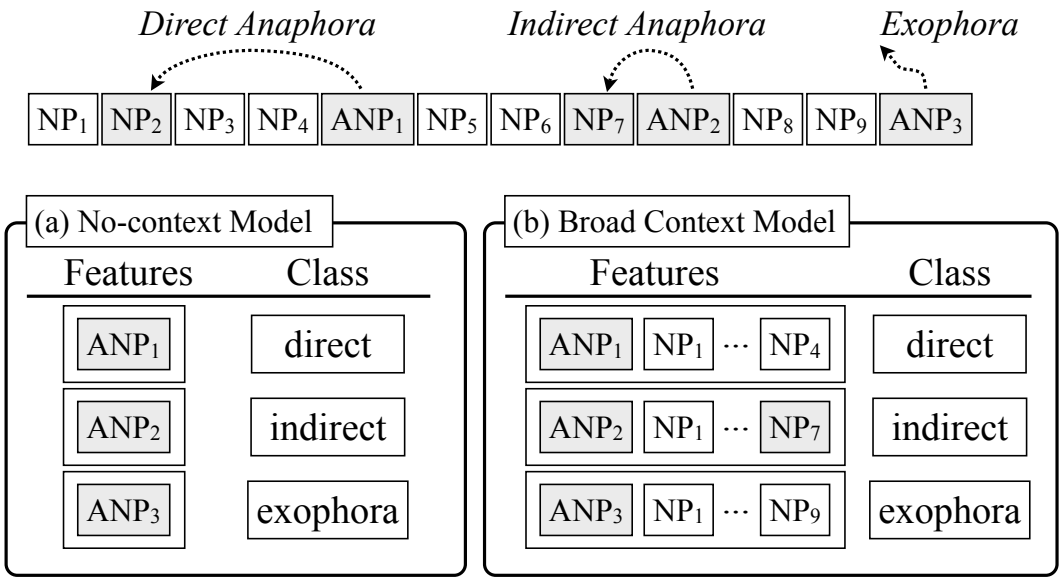


Figure 5.5: The procedure of training example generation for No-context model and Broad Context model.

Table 5.4: Feature set for No-context model and Broad Context Model

Feature	Description
ANAPHOR_DM_TYPE	Type of definiteness modifier of ANP. The possible value is one of "Kono", "Sono" or "Ano".
ANAPHOR_HEAD	Head morpheme of ANP.
ANAPHOR_POS	POS of ANP.
ANAPHOR_CASE	Case particle of ANP.
HOLDING_POS*	POS of all the candidates in the preceding sentences.
HAS_SYNONYM_OF_ANAPHOR*	1 if there exists a synonym of ANP in the preceding sentences; else 0.
HAS_HYPONYM_OF_ANAPHOR*	1 if there exists a hyponym of ANP in the preceding sentences; else 0.
HAS_STRING_MATCHED*	1 if there exists NP whose string matches the last string of (head of) ANP in the preceding sentences; else 0.
MAX_PMI*	Maximum PMI between ANP and each candidates in the preceding sentences.
MAX_NOUN_SIM*	Maximum noun-noun similarity between ANP and each candidates in the preceding sentences.

ANP denotes an anaphor. '*' denotes the features that capture the contextual information, which is only used for Broad Context model.

This model receives an anaphor and the most likely antecedent candidate(s) as its input. The classifier determines the anaphora type by utilizing information from both the anaphor and the selected candidate antecedent(s). This model has an advantage over the Broad Context model that it determines the anaphora type of a given anaphor taking into account the information of its most likely candidate antecedent. The most likely candidate can be expected to provide contextual information useful for anaphora type classification: for example, if *her new song* is selected as the best candidate antecedent in example (8), the anaphora type will be easily identified by using the lexical knowledge that *CD* is the semantically related object of *song*.

(8) *The artist announced her new song. I want to get the CD as soon as possible.*

Since we have 3 choices of the antecedent selection models as described in Section 5.3.1.1 (one is created from the mix strategy, and the rest is from the separate strategies), finally at least the following four models are available for anaphora type classification.

- *Mix Most Likely Antecedent Context (m-MLAC) Model*: Classifies anaphora type by using the information of the best candidate antecedent selected by the *mix antecedent selection model*.
- *Direct Most Likely Antecedent Context (d-MLAC) Model*: Classifies anaphora type by using the information of the most likely direct anaphoric antecedent selected by the *direct antecedent selection model*.
- *Indirect Most Likely Antecedent Context (i-MLAC) Model*: Analogous to the d-MLAC model, classifies anaphora type by using the information of the most likely indirect anaphoric antecedent selected by the *indirect antecedent selection model*.
- *Parallel Most Likely Antecedent Context (p-MLAC) Model*: Classifies anaphora type referring to two candidates selected by the *direct* and *indirect antecedent selection models*.

The p-MLAC model provides richer contextual information for classifying anaphora type than any other configuration because it can always refer to the most likely candidate antecedents of direct anaphora and indirect anaphora, which may be useful for determining anaphora type.

Training Procedure The training procedure of each model depends on which kinds of information is needed. Basically, we use a pair ⟨an anaphor, annotated antecedent or *pseudo*-antecedent⟩ as a training instance. It depends on the anaphora type of interested anaphor and the type of antecedent selection model that the classifier utilizes to determine whether it is annotated antecedent or *pseudo*-antecedent. We describe the training procedure of each model below using Figures 5.6, 5.7 and 5.8.

- *m-MLAC model*: Give an anaphor and an annotated antecedent with the label of anaphora type to the classifier, except an exophoric anaphor. Since the exophoric anaphor has no antecedent annotated in the training set, we pick up the *pseudo*-antecedent by using the *mix antecedent selection model* and give it to the classifier with the exophoric anaphor. For example, in Figure 5.6, a direct anaphor ANP_1 is paired with the annotated antecedent NP_2, and an indirect anaphor ANP_2 is paired with the annotated antecedent NP_7, and an exophoric anaphor ANP_3 is paired with *pseudo*-antecedent NP_5 selected by the *mix antecedent selection model*.

- *d-MLAC model*: Give an anaphor and an annotated antecedent with the label "direct" to the classifier if the anaphor is labeled as *direct anaphora*. In the case of an *indirect anaphoric* or *exophoric* anaphor, we select *pseudo*-antecedent by using the *direct antecedent selection model* and give it to the classifier, instead of using annotated antecedent even though the indirect anaphor has an annotated antecedent. Recall that the d-MLAC model determines an anaphora type by using the information of the most likely antecedent selected by the *direct antecedent selection model*. Figure 5.7 exemplifies this procedure. In Figure 5.7, a direct anaphoric anaphor ANP_1 is paired with the annotated antecedent NP_2, and an indirect anaphor ANP_2 is paired with the *pseudo*-antecedent NP_3 selected by the *direct antecedent*

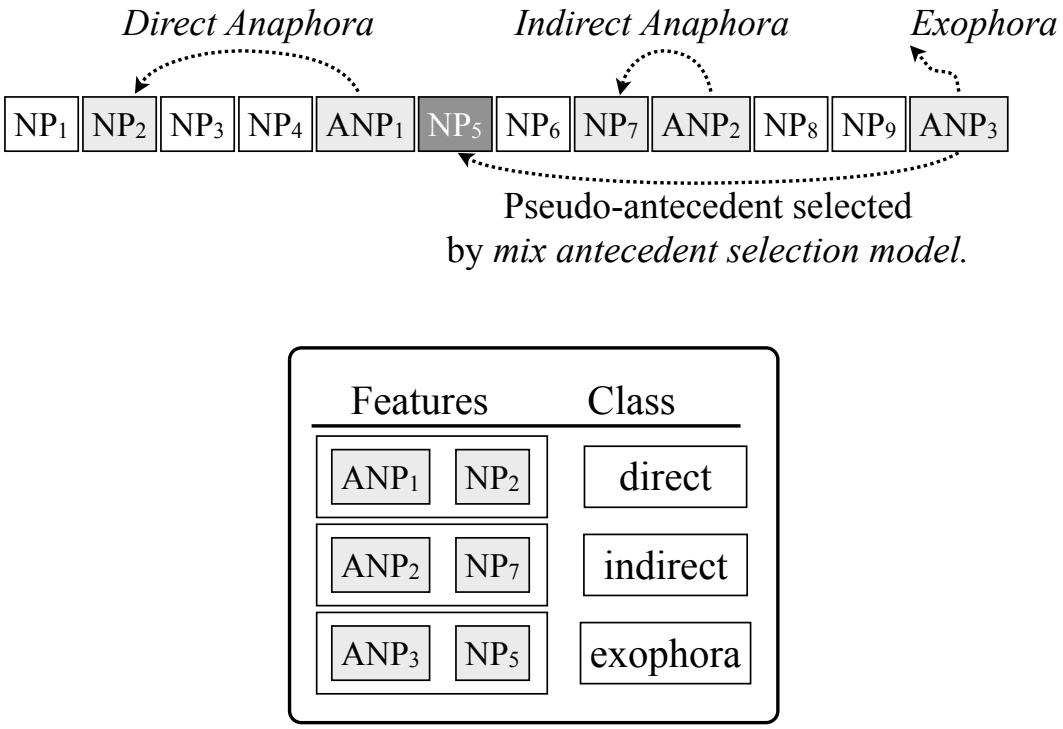


Figure 5.6: The procedure of training example generation for m-MLAC model.

selection model. Note that an annotated antecedent NP_7 is not used. Finally, an exophoric anaphor ANP_3 is paired with the *pseudo*-antecedent NP_5 selected by the *direct antecedent selection model*.

- *i-MLAC model*: Similarly to the d-MLAC model, give an anaphor and an annotated antecedent with a label "indirect" to the classifier if the *indirect* anaphor is given. In the case of a *direct* or *exophoric* anaphor, we select *pseudo*-antecedent by using the *indirect antecedent selection model* and give it to the classifier.

- *p-MLAC model*: Give a triplet ⟨an anaphor, a direct anaphoric (pseudo) antecedent, an indirect anaphoric (pseudo) antecedent⟩ to the classifier. It depends on the anaphora type how we give the two antecedents. First, in the case of a *direct* anaphor, we give an annotated antecedent and a *pseudo*-antecedent selected by the *indirect antecedent selection model* with a label "direct" to the classifier. Second, in the case of an *indirect* anaphor, we give an annotated antecedent and a *pseudo*-antecedent selected by the *direct antecedent selection model* with a label "indirect" to the classifier. Finally, for an exophoric anaphor, we give two *pseudo*-antecedents selected by the *direct antecedent selection model* and the *indirect antecedent selection model* with a label "exophora."

  We describe this procedure taking an example illustrated in Figure 5.8. For a direct anaphor ANP_1, we make the triplet by taking an annotated antecedent NP_2 as a direct anaphoric antecedent, and *pseudo*-antecedent NP_1 selected by the *indirect antecedent selection model* as an indirect anaphoric antecedent. Contrary to the *direct* anaphoric case, we take *pseudo*-antecedent NP_3 selected by the *direct antecedent selection model* as a direct anaphoric antecedent, and an annotated antecedent NP_7 as an indirect anaphoric antecedent in the case of an indirect anaphor ANP_2. For an *exophoric* anaphor ANP_3, we pick up two antecedents by using both the *direct* and *indirect antecedent selection model* since an exophoric anaphor has no annotated antecedent. Suppose NP_5 and NP_8 are selected respectively. Then we make the triplet from the two *pseudo*-antecedents; NP_5 and NP_8.

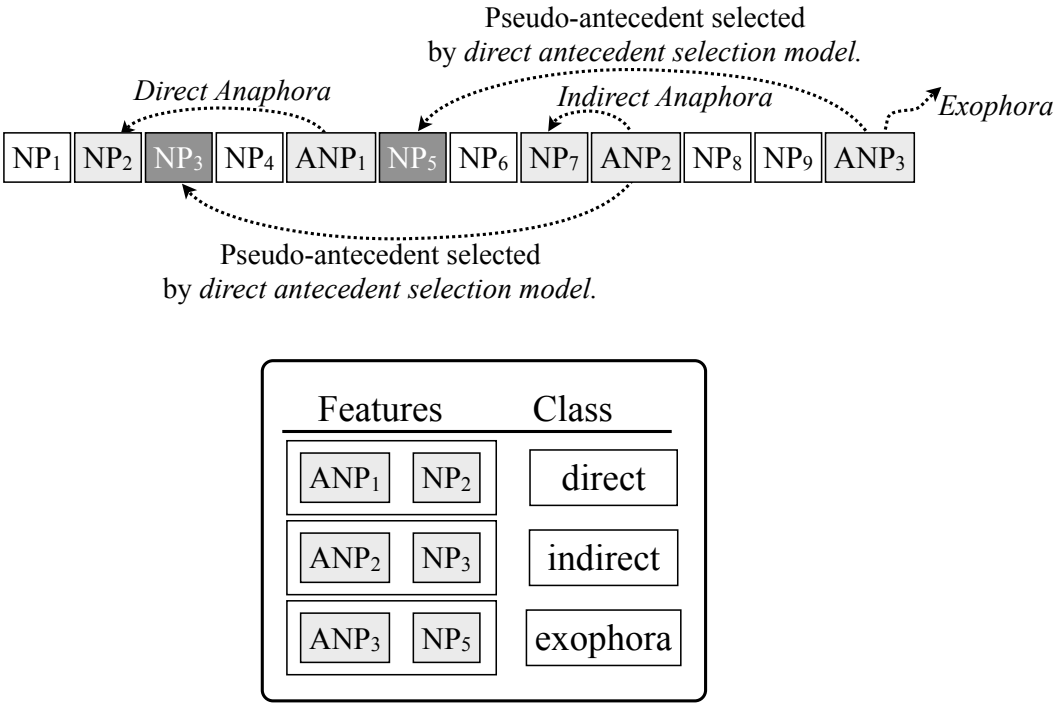


Figure 5.7: The procedure of training example generation for d-MLAC model.

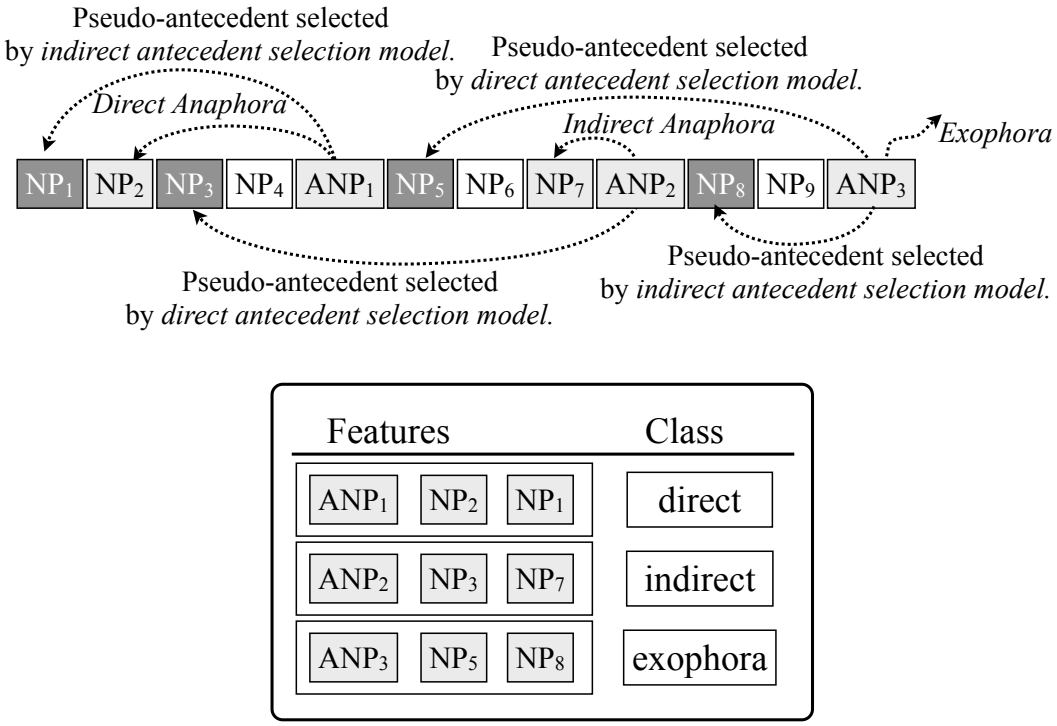


Figure 5.8: The procedure of training example generation for p-MLAC model.

Feature set The classifier uses the best candidate(s) antecedent selected by the antecedent selection model as its contextual information. This sort of information is encoded as features analogous to that for antecedent selection as summarized in Tables 5.1 and 5.2.

5.3.3 Anaphora Resolution Framework

We proposed three antecedent selection models (*mix, direct, and indirect antecedent selection model*) and four anaphora type classification models (No-context, Broad Context, and four Most Likely Antecedent Context models), each of which determines the referent and the anaphora type of a given anaphor. As mentioned in Section 1, one of the purposes of our work is to find an appropriate method of an anaphora resolution model which handles both the subtasks, i.e., antecedent selection and anaphora type classification. Thus, in this section, we integrate an antecedent selection model described in Section 5.3.1 with an anaphora type classification model described in Section 5.3.2 to find a practical anaphora resolution model capable of identifying a referent and resolving the ambiguity of an anaphora type in a real text (*issue 3*). In order to consider five anaphora resolution models here, we combine the antecedent selection models and anaphora type classification models described so far.

According to whether the antecedent selection is carried out before the anaphora type classification or after in a framework, we consider two types of configuration; Classify-then-Select (C/S) and Select-then-Classify (S/C) configuration. The C/S configuration first determines the anaphora type by using the No-context model or the Broad Context model, and selects an antecedent by using the direct or indirect antecedent selection models depending on the determined anaphora type. The S/C configuration, on the other hand, first selects an antecedent candidate by using mix, direct, or indirect antecedent selection model, and determines an anaphora type by using the information of the selected antecedent candidate. This configuration reselects an antecedent by the other antecedent selection model if necessary. We elaborate each configuration in Sections 5.3.3.1 and 5.3.3.2.

5.3.3.1 Classify-then-Select Configuration

Given an anaphor, this configuration first determines whether the anaphor bears either direct anaphora, indirect anaphora or exophora. If the anaphora type is judged as direct anaphora, then the direct antecedent selection model is called. If the anaphora type is judged as indirect anaphora, on the other hand, then the indirect antecedent selection model is called. There is no antecedent selection model called if exophora is selected.

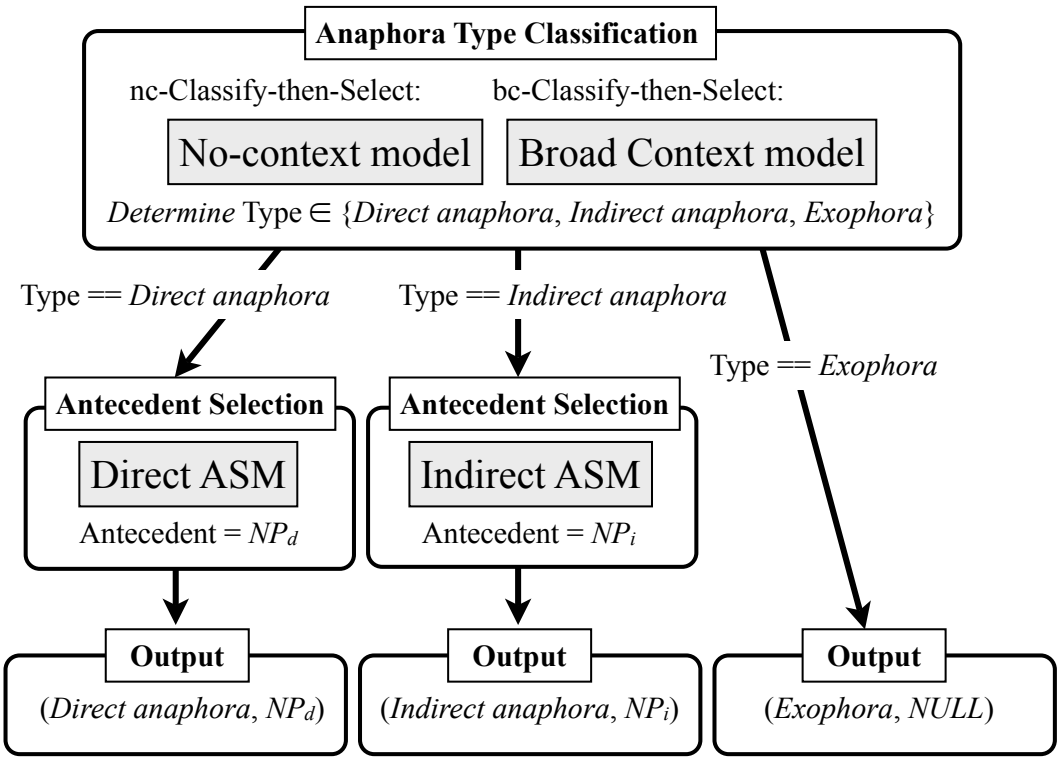
By altering the choice of anaphora type classification models, the following two alternative models are available for the Classify-then-Select configuration, each of which is illustrated in Figure 5.9.

- *nc-Classify-then-Select (ncC/S) Model*: Classify anaphora type of a given anaphor by using the No-context anaphora type classification model before selecting the antecedent.
- *bc-Classify-then-Select (bcC/S) Model*: Classify anaphora type of a given anaphor by using the Broad Context model before selecting the antecedent.

5.3.3.2 Select-then-Classify Configuration

Given an anaphor, this configuration first selects an antecedent candidate. Second, an anaphora type is determined by using the information of the candidate, i.e., this configuration determines an anaphora type by using the MLAC anaphora type classification model. In this section, we consider four models since we have alternative antecedent selection models and MLAC models.

- *m-Select-then-Classify (mS/C) Model*: Select an antecedent candidate with the *mix antecedent selection model*, and pass it to the m-MLAC model to classify an anaphora type. This model just returns the candidate passed to the m-MLAC model as the outputting antecedent. If the anaphor is classified as exophora, it outputs no antecedent. This procedure is illustrated in Figure 5.10.
- *d-Select-then-Classify (dS/C) Model*: Select an antecedent candidate by the *direct antecedent selection model* and then pass it to the d-MLAC model to



$*ASM$ denotes Antecedent Selection Model.

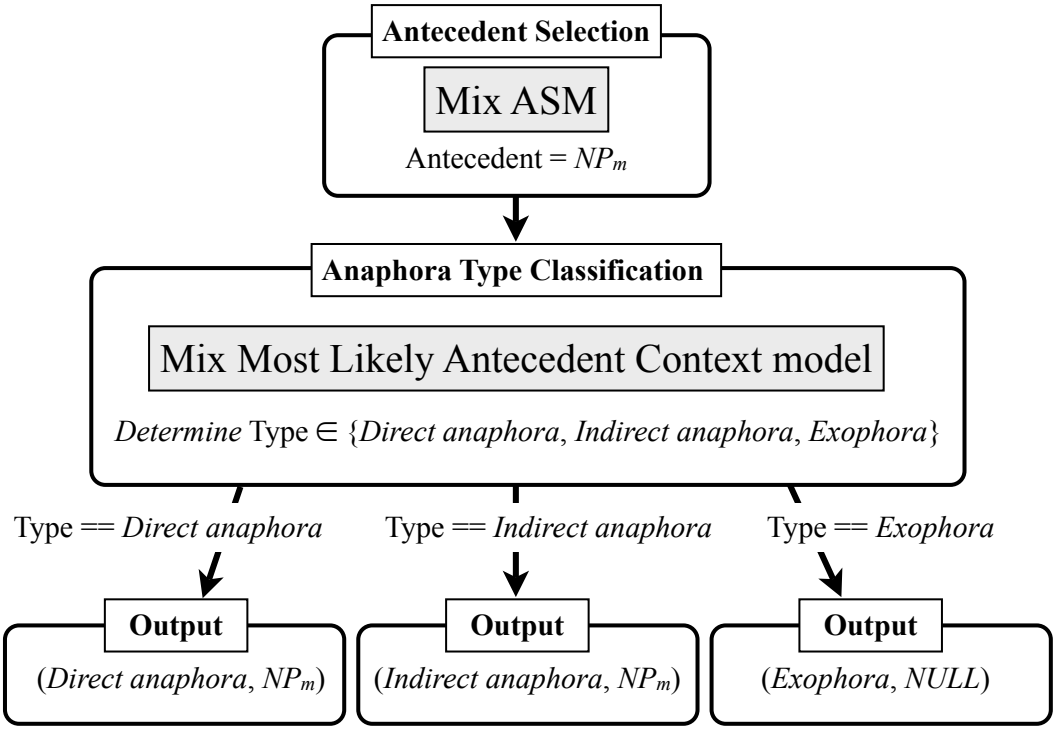
Figure 5.9: Classify-then-Select anaphora resolution framework.

classify the anaphora type. If the anaphor is classified as direct anaphora, it just returns the passed candidate as the outputting antecedent. If the anaphor is classified as indirect anaphora, search for the antecedent with the indirect antecedent selection model. It outputs no antecedent if the anaphor is classified as exophora. Figure 5.11 illustrates this procedure.

- *i-Select-then-Classify (iS/C) Model*: Select an antecedent candidate by the *indirect antecedent selection model* and then classify the anaphora type with the i-MLAC model. If the anaphor is classified as direct anaphora, search for the antecedent with the direct antecedent selection model. If the anaphor is classified as indirect anaphora, it just returns the candidate selected first as the outputting antecedent. It outputs no antecedent if the anaphor is classified as exophora. This procedure is shown in Figure 5.12.

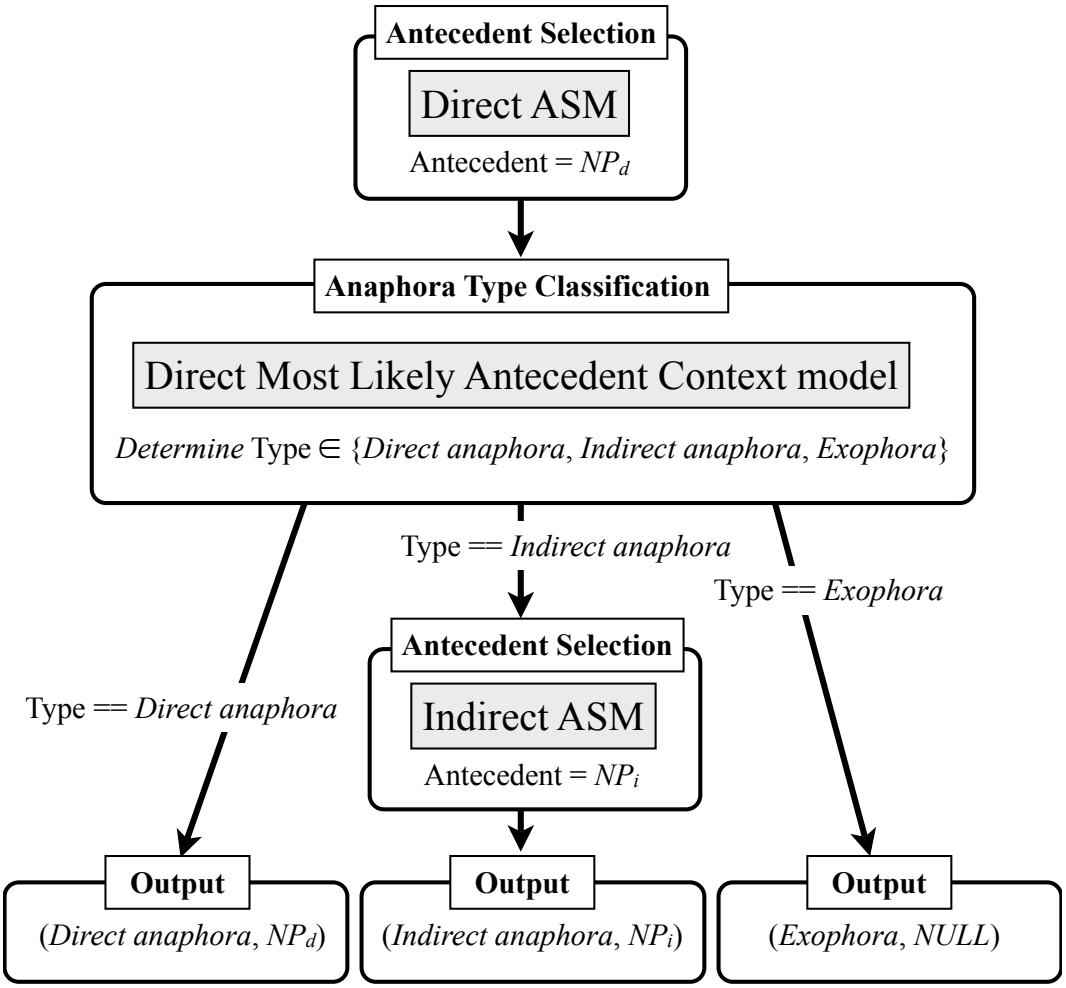
- *p-Select-then-Classify (pS/C) Model*: Select two antecedent candidates by the *direct* and *indirect antecedent selection models* in parallel, and then pass both the candidates to classify the p-MLAC model to determine the anaphora type. If the anaphor is classified as direct anaphora, it outputs an antecedent selected by the *direct antecedent selection model*. If the anaphor is classified as an indirect anaphora, it outputs the antecedent selected by the *indirect antecedent selection model*. If the anaphor is classified as exophora, it outputs no antecedent. Figure 5.13 illustrates this procedure.

As mentioned in Section 5.3.2.3, note that this configuration is expected to have an advantage over the C/S configuration in that it determines an anaphora type taking into account the information of its most likely antecedent candidate, instead of all the candidates. It may do harm to an anaphora type classification to use the information of all the candidates, since it includes too much information useless or harmful to classify an anaphora type.



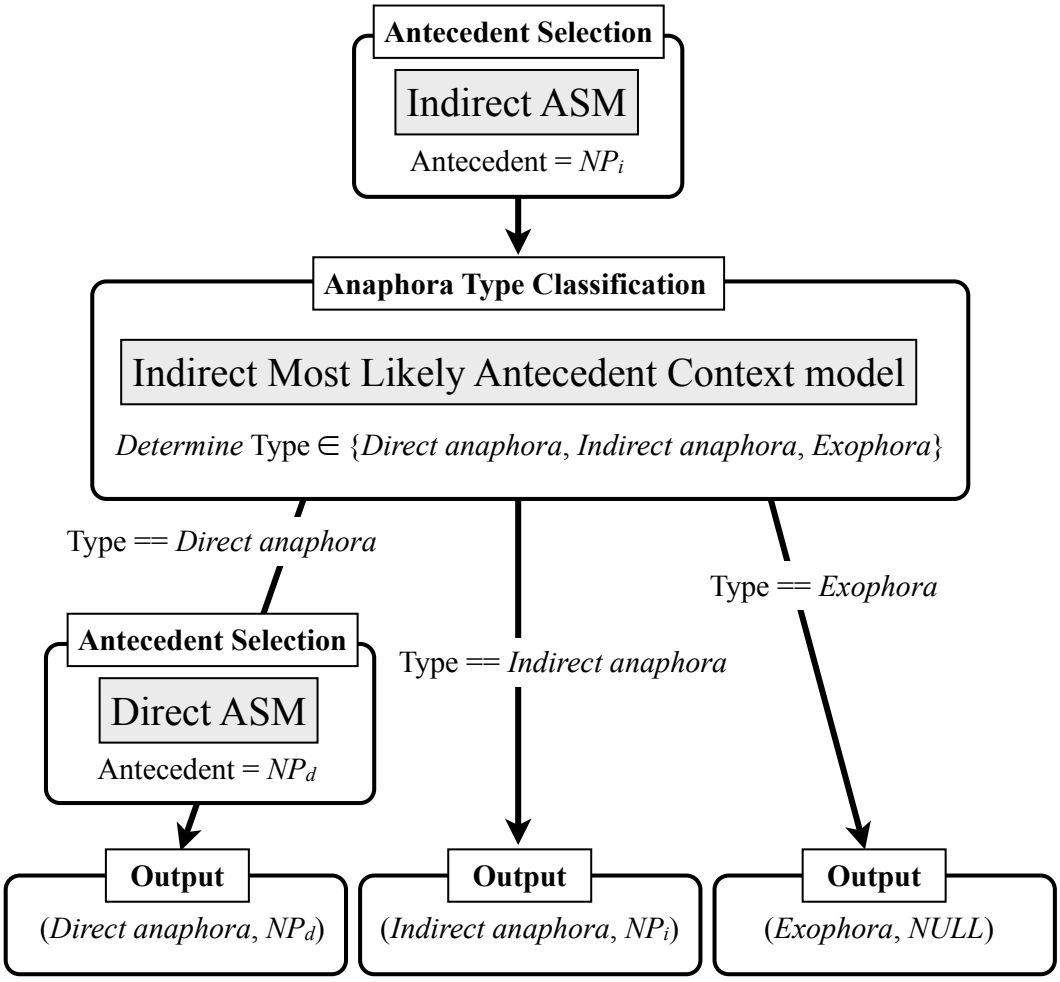
$*ASM$ denotes Antecedent Selection Model.

Figure 5.10: m-Select-then-Classify anaphora resolution framework.



$*ASM$ denotes Antecedent Selection Model.

Figure 5.11: d-Select-then-Classify anaphora resolution framework.



$*ASM$ denotes Antecedent Selection Model.

Figure 5.12: i-Select-then-Classify anaphora resolution framework.

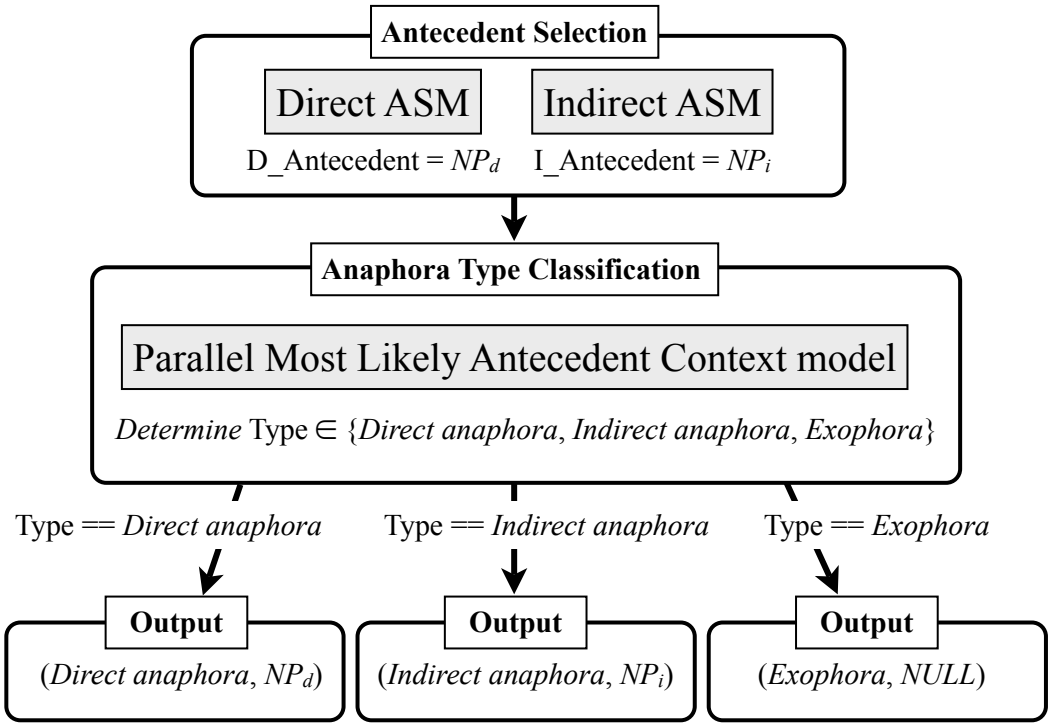


Figure 5.13: p-Select-then-Classify anaphora resolution framework.

ASM denotes Antecedent Selection Model.

Table 5.5: Distribution of anaphoric relations in the broadcast articles.

Syntax	Direct	Indirect	Exophora	Ambiguous
Noun	530	466	-	0
Predicate	70	435	-	8
Overall	600	901	248	8

'Noun' and 'Predicate' denote the syntactic category of an antecedent. 'Ambiguous' was annotated to an anaphor which holds both direct and indirect anaphoric relations. In our evaluations, we discarded such instances.

Table 5.6: Distribution of anaphoric relations in the editorial articles.

Syntax	Direct	Indirect	Exophora	Ambiguous
Noun	550	561	-	0
Predicate	114	883	-	2
Overall	664	1,444	222	2

The definition of 'Noun', 'Predicate' and 'Ambiguous' follows Table 5.5.

5.4 Dataset

For training and testing our models, we created an annotated corpus that contains 2,929 newspaper articles consisting of 19,669 sentences for 2,320 broadcasts, 18,714 sentences for 609 editorials, which are the same articles as in the NAIST Text Corpus [Iida et al., 2007]. The NAIST Text Corpus also contains anaphoric relations of noun phrases, but they are strictly restricted as coreference relations (i.e. two NPs must refer to the same entity in the world). For this reason, most NPs marked with a definiteness modifier that we need are not annotated even when two NPs have a direct-anaphoric relation. Therefore, we re-annotated (i) direct anaphoric relations, (ii) indirect anaphoric relations and (iii) exophoric noun phrases of noun phrases marked by one of the three definiteness modifiers, that is *this* (この), *the* (その), and *that* (あの). In the specification of our corpus, not only noun phrases but verb phrases are chosen as antecedents. For example, the verbal predicate $calculates_{(i')}$ is selected as an antecedent of *the* $precomputation_{(i)}$ in example (9).

(9) システムは前もって値を計算する$_{(i')}$。その前計算$_{(i)}$はシステムの性能を大幅に向上させている。
The system calculates$_{(i')}$ the value in advance. The precomputation$_{(i)}$ significantly improves its performance.

We also annotated anaphoric relations in the case where an anaphor is anaphoric with more than two antecedents. For example, we label indirect anaphoric relations for the two pairs of NPs *mouse devices–the other items* and *keyboards–the other items* as seen in example (10).

(10) ABC コンピュータはマウス$_{(i')}$とキーボード$_{(j')}$の値下げを発表した。
その他の商品$_{(i,j)}$については値下げをしないと主張した。
ABC computer announced that they reduced the price of mouse devices$_{(i')}$ and keyboards$_{(j')}$.
They claimed that they would not cut the price of the other items$_{(i,j)}$.

Finally, we obtained 1,264 instances of direct anaphora, 2,345 instances of indirect anaphora, and 470 instances of exophora. The detailed statistics are shown in Tables 5.5 and 5.6. To assess the reliability of the annotation, we estimated its agreement rate with the two annotators from 418 examples[1] in terms of K statistics [Sidney and Castellan, 1988]. It resulted in $K = 0.73$, which indicates good reliability. For measuring the agreement ratio of antecedent selection, we used 322 examples (109 for direct anaphora and 213 for indirect anaphora) whose anaphora types are identically identified by both two annotators. The agreement ratio was calculated[2] according to the following equation:

$$Agreement = \frac{\#\text{ of instances which both two annotators identified the same antecedent}}{\#\text{ of all instances}}.$$

The agreement ratio for annotating direct-anaphoric relation obtained 80.7% (88/109). However, for 21 examples whose antecedents are not identically selected

[1]These examples are randomly sampled from our corpus, and account for 10% of all the examples.

[2] We regarded the matching of the rightmost offset as the agreement. When multiple antecedents are annotated, the criterion of matching is that one of the antecedents is at least identical with one of the antecedents annotated by the other annotator.

by the annotators, our analysis revealed that 52.4% (11/21) of these examples are cases where the antecedents annotated by the two annotators are different but in anaphoric relation, which should be regarded as an agreement. Therefore, the inter-annotator agreement ratio of direct-anaphoric relation achieves 90.8% (99/109), which indicates good reliability but it is required to consider anaphoric chains in the annotation procedure. The agreement ratio of indirect-anaphoric relation, on the other hand, obtained a comparatively lower ratio of 62.9% (134/213). One of the typically non-matching cases is shown in example (11).

(11) 政府$_{(i)}$は明日までに委員$_{(j)}$を決める方針だ。その人選$_{(k)}$は我々にも影響が及ぶだろう。
The government$_{(i)}$ is going to determine the member of the committee$_{(j)}$ by tomorrow. Probably the election$_{(k)}$ will also affect us.

In this example, both *the government* and *the member of the committee* are considered to be associated objects of *the election*, which indicates that multiple discourse elements are often associated with one anaphor in various semantic relations in indirect anaphora. We should reflect on such problems when the annotation scheme and task definition of indirect anaphora resolution are argued, including bridging reference resolution.

5.5 Experiments

We conduct empirical evaluations in order to investigate the three issues shown in Section 1. First, we compare two antecedent selection models, the single and separate models described in Section 5.3.1 in order to find out *issue 1*, i.e., whether an antecedent selection model should be trained separately for direct anaphora and indirect anaphora. Second, the anaphora type classification models described in Section 5.3.2 are evaluated to explore what information helps with the anaphora type classification (*issue 2*). Finally, we evaluate the overall accuracy of the entire anaphora resolution task to explore how the models can be best configured (*issue 3*).

In our experiments, we used anaphors whose antecedent is a head of an NP that appears in the preceding context of the anaphor (i.e., cataphora is ignored), only taking articles in the broadcast domain into account. Therefore, we used 572 instances of direct anaphora, 878 instances of indirect anaphora and 248 instances of exophora. The evaluation was carried out by 10-fold cross-validation. In our evaluation of antecedent selection, if a selected antecedent is in the same direct-anaphoric chain as the labeled antecedent, this selected antecedent is evaluated as correct[1].

For creating binary classifiers used in antecedent selection and anaphora type classification, we adopted Support Vector Machines [Vapnik, 1995b][2], with a polynomial kernel of degree 2 and its default parameters.

We adopt the *one-versus-rest* method for the three-way classification for anaphora types. In other words, we recast the multi-class classification problem as combinations of a binary classification. Given an anaphor, each anaphora type classifier outputs a score that represents the likelihood of its anaphora type. According to these three scores, we select the anaphora type that achieves the maximum score.

5.5.1 Results of Antecedent Selection

The results of antecedent selection are shown in Table 5.7. The results[3] indicate that the Separate Strategy outperforms the Mix Strategy on two anaphora types. As for *issue 1*, we conclude that the information used for antecedent selection should be separated for each anaphora type and the selection models should be trained for each anaphora type. We therefore discard the mix strategy for the further experiments (i.e. discarding the m-MLAC model and the mS/C model).

We also illustrate the learning curves of each model, shown in Figure 5.14. Reducing the training data to 50%, 25%, 12.5%, 6.25% and 3.13%, we conducted the evaluation over three random trials for each size and averaged the accuracies.

[1]We manually checked our results because of the lack of annotation of anaphoric chains as noted in Section 5.4. Due to the cost of this manual checking, we took only the broadcast articles into account in our experiments, leaving the editorials out.

[2]SVM^{light} http://svmlight.joachims.org/

[3]The accuracy of the separate strategy is better than the mix strategy with statistical significance ($p < 0.01$, McNemar test).

Table 5.7: Results of antecedent selection

Anaphora Type	Mix Strategy	Separate Strategy
Direct anaphora	63.3% (362/572)	65.4% (374/572)
Indirect anaphora	50.5% (443/878)	53.2% (467/878)
Overall	55.2% (801/1,450)	**58.0% (841/1,450)**

Table 5.8: Precision, recall and F-value of anaphora type classification.

	Direct Anaphora			Indirect Anaphora			Exophora		
Model	P	R	F	P	R	F	P	R	F
NC	67.7%	74.5%	70.9%	80.6%	87.1%	83.7%	75.0%	36.3%	48.9%
BC	69.4%	73.4%	71.4%	74.9%	87.5%	80.7%	92.5%	25.0%	39.4%
d-MLAC	70.9%	84.6%	**77.1%**	83.2%	85.6%	**84.4%**	90.1%	40.3%	55.7%
i-MLAC	67.7%	74.8%	71.1%	78.1%	88.3%	82.9%	93.2%	27.8%	42.9%
p-MLAC	71.2%	82.0%	76.1%	82.1%	86.7%	84.3%	91.9%	41.1%	**57.2**%

Figure 5.14 indicates that in the direct antecedent selection model the accuracy becomes better as the training data increase, whereas the increase of the indirect one looks difficult to improve although our data set included more instances for indirect anaphora than for the direct one. These results support the finding in previous work that an indirect anaphora is harder to resolve than direct anaphora and suggest that we need a more sophisticated antecedent selection model for indirect anaphora.

5.5.2 Results of Anaphora Type Classification

Now, we move on to *issue 2*. The results of anaphora type classification are shown in Tables 5.8 and 5.9. The BC model obtained the lowest accuracy of 73.6%, which indicates that contextual information features proposed in the literature [Vieira and Poesio, 2000, etc.], such as HAS_STRING_MATCHED, were not actually informative. Note that the performance of the BC model is lower than the NC model[1], which identifies an anaphora type by using only the information of an anaphor. On the other hand, the d-MLAC model successfully improved its performance by using the information of selected candidate antecedent as

[1]The difference is statistically significant ($p < 0.06$, McNemar test).

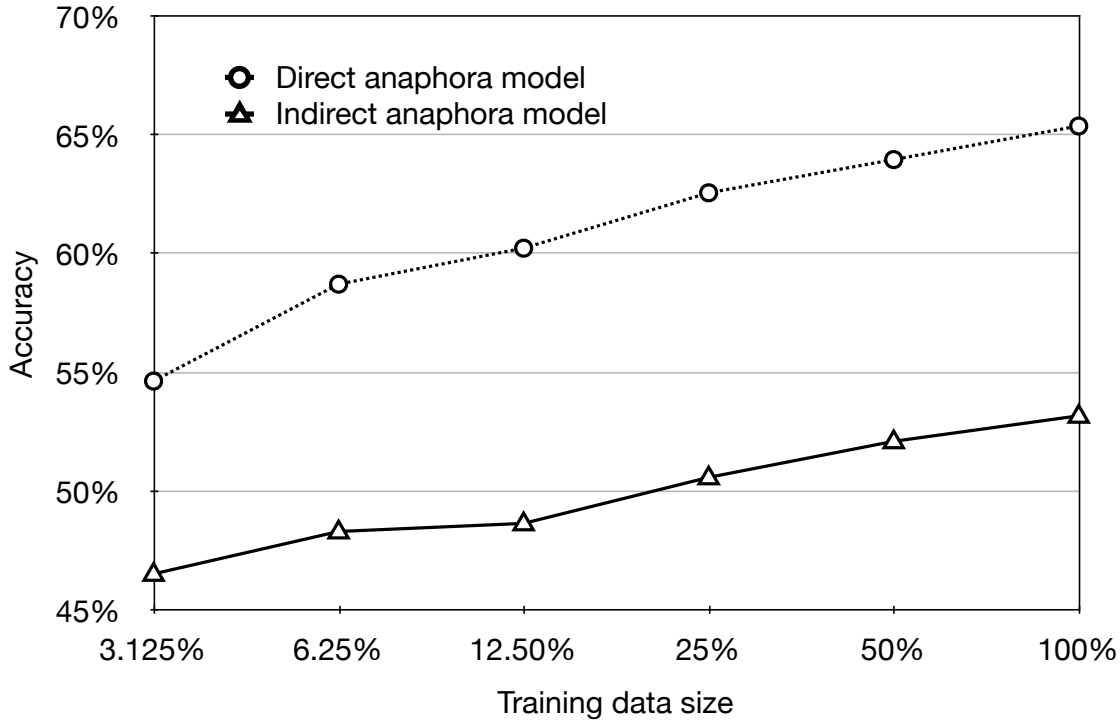


Figure 5.14: Learning curve for separate models.

the contextual information. The d-MLAC model achieved the best accuracy of 78.7%, which indicates that the selected best candidate antecedent provides useful contextual information for anaphora type classification[1]. The i-MLAC and p-MLAC models, however, do not improve their performance as well as the d-MLAC model although it uses the selected best candidate(s) information. It is considered that the fundamental reason is the poor performance of the *indirect* antecedent selection model as shown in Table 5.7, i.e., the *indirect* antecedent selection model does not provide correct contextual information to anaphora type classification. It is expected that all the MLAC models get better performance when the antecedent selection model improves.

Table 5.9: Accuracy of anaphora type classification.

Model	Accuracy
NC	75.4%
BC	73.6%
d-MLAC	**78.7%**
i-MLAC	74.9%
p-MLAC	78.4%

Table 5.10: Overall results of anaphora resolution

Model	Accuracy
nc-Classify-then-Select	47.3% (803/1,698)
bc-Classify-then-Select	46.3% (787/1,698)
d-Select-then-Classify	**50.6% (859/1,698)**
i-Select-then-Classify	46.3% (787/1,648)
p-Select-then-Classify	50.4% (855/1,698)

5.5.3 Results of Overall Anaphora Resolution

Finally, we evaluated the overall accuracy of the entire anaphora resolution task given by:

$$Accuracy = \frac{\text{\# of instances whose antecedent and anaphora type is identified correctly}}{\text{\# of all instances}}.$$

The results are shown in Table 5.10. The dS/C model achieved the best accuracy, which is significantly better than the Classify-then-Select models. As for *issue 3*, we found that it is the best configuration that it selects an antecedent first, and then passes the antecedent to an anaphora type classifier to determine the anaphora type.

5.5.4 Error Analysis of Antecedent Selection

Our error analysis revealed that a majority (about 60%) of errors in direct anaphora were caused by the fact that both correct and incorrect candidates

[1]The d-MLAC model outperformed the NC, BC models with statistical significance using $p < 0.03$, $p < 0.01$, as McNemar test parameters respectively.

belong to the same semantic category. Example (12) shows a typical selection error:

(12) 私は映画$_{(j)}$の知識がないが、『フランケンシュタイン』$_{(i')}$ぐらいは知っている。この映画$_{(i)}$は、本当に名作だ。
I don't have good knowledge of movies$_{(j)}$ but still know of "Frankenstein"$_{(i')}$. I think this movie$_{(i)}$ is indeed a great masterpiece.

where the wrong candidate "映画$_{(j)}$ (*movies*$_{(j)}$)" was selected as the antecedent of "この映画$_{(i)}$ (*this movie*$_{(i)}$)".[1] As can be imagined from this example, there is still room for improvement by carefully taking into account this kind of error using other clues such as information from salience.

For indirect anaphora, we analyzed our resource to capture the associativeness between an anaphor and its antecedent, encoded as Pmi in the feature set. Our analysis indicated that about half of the pattern '*ANT* of *ANP*', which occurred in the test data, had been assigned a minus value, i.e., no positive association found between an anaphor and its antecedent for the resource when applying Pmi. To evaluate the contribution to our model, we conducted an evaluation where the Pmi feature set was disabled. As a result of this additional evaluation, the model obtained 51.4% (451/878), which is no significant difference compared with the original accuracy. We need to find more useful clues to capture the associativeness between an anaphor and the related object in indirect anaphora. The low quality of our annotating data of indirect-anaphoric relation, as mentioned in Section 5.4, might be also one of the reasons for the low accuracy of indirect anaphora resolution.

5.5.5 Error Analysis of Anaphora Type Classification

The identification of exophora is a more difficult task than the other anaphora types as shown in the low F-measure and recall in Table 5.8. Our analysis for the exophoric instances misclassified by the d-MLAC model revealed that the typical errors were temporal expressions such as 年 (*year*), 日 (*day*) and 時期 (*period*). We observed that such expressions occurred as not only exophora but also as the

[1] In Japanese, the plural form of a noun is not morphologically distinguished from its singular form.

Table 5.11: The majority of misclassified-exophoric instances

	Occurrences in our corpus		
NP of an anaphor	Direct anaphora	Indirect anaphora	Exophora
年 (*year*)	42.9% (9/21)	9.5% (2/21)	47.6% (10/21)
日 (*day*)	68.3% (82/120)	0.9% (1/120)	30.8% (37/120)
時 (*time*)	8.9% (5/56)	82.1% (46/56)	8.9% (5/56)
時期 (*period*)	25.0% (5/20)	35.0% (7/20)	40.0% (8/20)

other anaphora types many times, as summarized in Table 5.11, which indicates that the interpretation of temporal expression is also important for identifying the other anaphora types. In our current framework, however, it is hard to recognize such expressions accurately since the precise recognition of temporal expressions is required to identify a relation between an event specified by the expression and the other events. We consider integrating the framework of temporal relation identification, which has been proposed in the evaluation-oriented studies such as TempEval[1], with anaphora type classification framework, which will be our future work.

5.6 Conclusion

We have addressed the three issues of nominal anaphora resolution for Japanese NPs marked by a definiteness modifier under two subtasks, i.e., *antecedent selection* and *anaphora type classification*. The issues we addressed were: (i) how the antecedent selection model should be designed, (ii) what information helps anaphora type classification, and (iii) how the antecedent selection and anaphora type classification should be carried out. Our empirical evaluations showed that the separate strategy achieved better accuracy than the mix strategy for antecedent selection, and the d-MLAC model gives the best result for anaphora type classification. As for the integrated models, the d-Select-then-Classify model achieved the best accuracy. We have made several findings through the evaluations: (i) an antecedent selection model should be trained separately for each anaphora type using the information useful for identifying its antecedent, (ii)

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

the best candidate antecedent selected by an antecedent selection model provides contextual information useful for anaphora type classification, and (iii) the antecedent selection should be carried out before anaphora type classification.

However, there is still considerable room for improvement in both subtasks. Our error analysis for antecedent selection reveals that the wrong antecedent, which belongs to the same semantic category as correct antecedent, is likely to be selected while selecting direct-anaphoric antecedent, and the association measure of indirect-anaphoric relatedness does not contribute to selecting the indirect-anaphoric antecedent. For anaphora type classification, our analysis reveals that temporal expressions typically cause error in the identification of exophora. To recognize such expressions precisely, we will consider integrating temporal relation identification with anaphora type classification. Our future work also includes taking general noun phrases into account in anaphora resolution. In the next chapter, we propose the inference-based approach to anaphora resolution, which overcomes these limitations.

Chapter 6

Inference-based Approach to Coreference Resolution

In this chapter, we propose an inference-based direct anaphora resolution model. Particularly, we focus on *coreference resolution* problem, where we need to identify a set of mentions that refers to the same entity in the world. Conventional approaches to coreference resolution have exploited world knowledge to capture syntactic or semantic compatibility between mentions, encoding them as a feature vector for machine learning-based classifiers. However, as mentioned in Chapter 6, there exist many cases where several antecedents are syntactically or semantically compatible with an anaphoric expression, and therefore the existing approaches are not guaranteed to identify correct antecedents in such cases. Following Rahman and Ng [2012], we refer to these cases as *difficult coreference problems*.

In this chapter, to remedy this problem, we propose a machine learning-based hybrid model that combines the conventional compatibility-based approach with a logical inference-based approach. Our key idea is that the information of implicit events inferred by logical inference (henceforth, *implicit events*) provides useful clues for selecting the correct antecedent. We integrate those two approaches to complement the weakness of each approach, using an abductive inference framework.

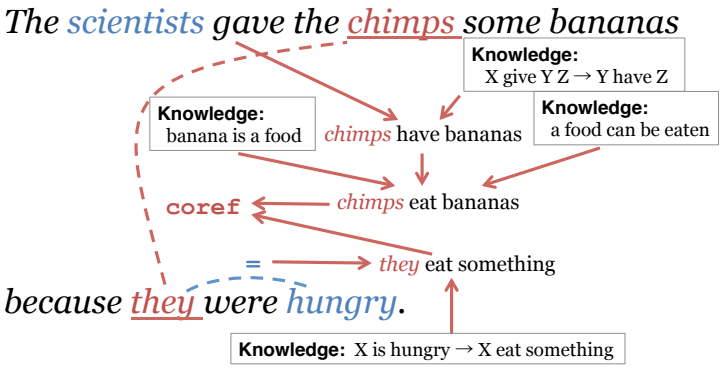


Figure 6.1: Example of inference-based coreference resolution.

6.1 Motivation

An inference-based formulation is appealing for coreference resolution because it is a realization of the observation that we understand new material by linking it with what we already know. It instantiates in natural language understanding the more general principle that we understand our environment by coming up with the best explanation for the observables in the environment.

Hobbs et al. [1993] show that the lowest-cost abductive proof provides the solution to a whole range of natural language pragmatics problems, such as word sense disambiguation, anaphora and metonymy resolution, interpretation of noun compounds and prepositional phrases, detection of discourse relations, etc. For examples of application of abduction to discourse processing see [Charniak and Goldman, 1991; Inoue and Inui, 2011; Ovchinnikova, 2012; Ovchinnikova et al., 2011].

Let us elaborate how this idea helps us to solve difficult coreference problems with the following example sentence:

- *The scientists gave the chimps some bananas because they were hungry.*

To help the readers follow our discussion easily, we describe our idea using the diagram of inference flow in Figure 6.1.[1] In Figure 6.1, we infer that *[the chimps]$_j$ would have the bananas*, applying the causal knowledge that giving causes having

[1]The arrows in the diagram are *not* logical implications.

to the observation that the scientists gave some bananas to them. We also infer that *the chimps would eat the bananas* since having bananas causes the desire of eating. On the other hand, we infer that *[they]$_j$ would eat something*, because being hungry causes eating something. Notice that we have two *eating* events that are derivable from the observed text.

Following the assumption above, we conclude that two *eating* events are likely to be coreferent; that is, [they]$_j$ and [chimps]$_j$ should be coreferent. Although both *the scientists* and *the chimps* are semantically compatible with [they]$_j$, capturing coreference relation between implicit *eating* events provides the clue that supports *the chimps* is a better antecedent for *they*.

In order to create a computational mechanism that realizes these procedures, we need at least two subtasks: (i) deriving plausible implicit events from observed information (*implicit event derivation*), and (ii) resolving coreference between observed mentions (*coreference resolution*), exploiting the derived implicit information as a clue. In this chapter, we recast the two subtasks as the problem of abductive explanation finding and then define a trainable score function that evaluates the abductive explanation in terms of the goodness of coreference relations and the reliability of inference.

6.2 The Model

6.2.1 Generation of Abductive Explanation for Implicit Event Derivation and Coreference Resolution

Following Hobbs et al. [1993], we jointly model the task of implicit event derivation and coreference resolution as the abductive inference problem, where a target text and world knowledge are regarded as the observation and the background knowledge respectively. We recall that the spirit of Hobbs et al. [1993] is that the process of interpreting sentences is reducible to the process of finding the minimal explanation to the sentences; that is, the process of natural language understanding amounts to performing abductive inference, where the observation is the logical forms (LFs) of a target discourse, and the background knowledge is a set of LFs of inference rules derived from world knowledge (or, could be meta-

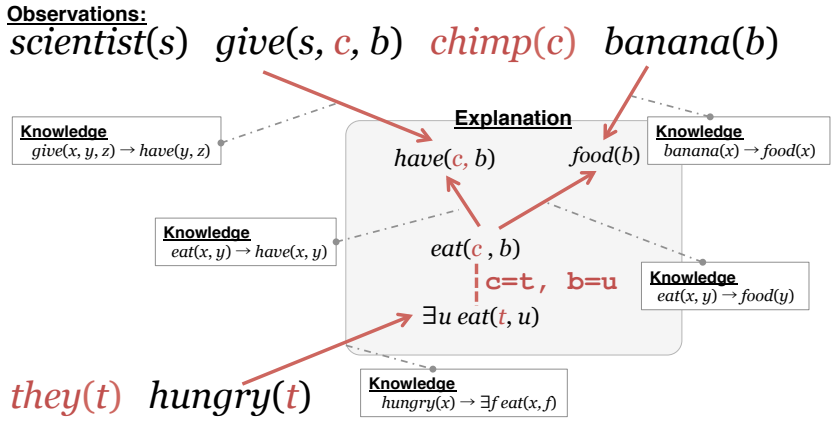


Figure 6.2: Example of abduction-based coreference resolution.

level knowledge). The LFs in each candidate explanation represents one possible interpretation of the text.

In our context, each possible combination of decisions about derivation of implicit events and coreference are mapped into the LFs in each abductive explanation. The motivation for using an abductive inference framework is that we resort to the minimality of explanation for evaluating the plausibility of inferred implicit events.

More formally, given a target text and world knowledge as the observation O and background knowledge B, we find the best explanation $\hat{H}$:

$$\hat{H} = \arg\max_{H \in \mathcal{H}_{O,B}} score(H), \tag{6.1}$$

where each explanation $H \in \mathcal{H}_{O,B}$ includes the decision about implicit events, or coreference relations, and $score(H)$ is a score function that jointly evaluates the goodness of coreference decisions and derivation of implicit events in H.

In the following, we summarize the mappings between natural language and LFs used in abduction-based coreference resolution. Figure 6.2 illustrates the example abductive inference to Figure 6.1 using these mappings.

- A target text: observation (e.g. $\exists c, t(chimp(c) \wedge scientist(s) \wedge they(t))$)
  - Mentions: logical variables (e.g. c, t, s)

- Event: literals (e.g. $eat(c)$)

- World knowledge: background knowledge (e.g. $\forall x(eat(x) \rightarrow hungry(x))$)

- Output: Explanation
  - Coreference: equality assumptions (e.g. $t = s, c = t$)
  - Event coreference: unification of two literals (e.g. $\{eat(c), eat(t)\} \rightarrow \{eat(c), c = t\}$)
  - Implicit events: literals derived by backward inference (e.g. $[(\forall x eat(x) \rightarrow hungry(x)) \wedge hungry(c)] \rightarrow eat(c)$)

6.2.2 Scoring Plausibility of Abductive Explanations

Now we move on to the issue of how to design the abductive score function $score(H)$ in equation (6.1). How can we say that one explanation is better than the others in our context? Since our model jointly infers implicit events and coreference relations, the score function should be capable of evaluating abductive explanations in terms of two aspects: (i) the goodness of coreference relations and (ii) the goodness of inference used for deriving implicit events. The second aspect is needed, because abductive inference is not always valid, unlike deductive inference.

To take the two aspects into account, we first model the score function as a linear model, and then encode these information in the feature function. Let $\mathbf{\Phi}(H) = \{\phi_1(H), \phi_2(H), ..., \phi_n(H)\}$ be a n-dimensional feature vector of an explanation H, and $\mathbf{w} = \{w_1, w_2, ..., w_n\}$ be a n-dimensional weight vector. We then define the score function as follows:

$$score(H; \mathbf{w}) = \mathbf{w} \cdot \mathbf{\Phi}(H) = \sum_{i=1}^{n} w_i \cdot \phi_i(H) \tag{6.2}$$

We refer to $\mathbf{w}$ as the *parameter* of score function. In the rest of this section, we decompose the feature vector $\mathbf{\Phi}$ into two parts $\mathbf{\Phi}_C, \mathbf{\Phi}_I$, namely the feature vector for coreference decisions, and the feature vector for implicit event derivation

respectively. For inference and learning, we use the proposed method described in Chapter 4.

Coreference: We first describe a score function for coreference evaluation. As a coreference resolution model, we use clustering-based approach [McCallum and Wellner, 2004], which has several advantages to traditional pairwise-mention or entity-mention approaches. Because the clustering-based approach globally evaluates the overall coreference relations, it does not lead to globally inconsistent coreference relation decision such as (Obama–He) and (Obama–She).

As mentioned earlier, each decision about coreference relation corresponds to each equality assumption. In order to implement clustering-based coreference model, we sum up the goodness of equality assumptions included in an explanation as follows:

$$\mathbf{\Phi}(H) = \sum_{(x,y)\in eqs(H)} \mathbf{\Phi}_C(x, y, O), \tag{6.3}$$

where $eqs(H)$ is a set of equality assumptions in H, and O is an observation. The feature function $\mathbf{\Phi}_C$ models the semantic compatibility between two mentions x, y based on the observed information (see Sec. 6.4.1). The transitive relations over equality assumptions are guaranteed by the axioms of equality in first-order logic. Finding the best explanation that maximizes this score function amounts to correlation clustering of mentions [Finley and Joachims, 2005].

Modeling implicit event derivation: We then describe our full model. We extend equation (6.3) to evaluate the likelihood of both coreference resolution and implicit event derivation. First, we replace $\mathbf{\Phi}_C(x, y, O)$ with $\mathbf{\Phi}_C(x, y, H)$ to use the information that is derived by abductive inference. Second, we add two new terms to take the plausibility of inferred implicit events into account.

For evaluating the likelihood of implicit event derivation, we resort to the minimality of explanation, following [Hobbs et al., 1993]. Intuitively, the score function gives a penalty for assuming specific and unreliable information but rewards for explaining other information or inferring the same information from different observations. We model this intuition by modeling the feature function

with two terms: (i) what axioms are used for construction of H ($\mathbf{\Phi}_A$), and (ii) what literals are explained in H ($\mathbf{\Phi}_L$):

$$\mathbf{\Phi}(H) = \sum_{(x,y)\in eqs(H)} \mathbf{\Phi}_C(x, y, H) + \sum_{a\in axioms(H)} \mathbf{\Phi}_A(a) + \sum_{p\in literals(H)} \mathbf{\Phi}_L(p), \tag{6.4}$$

where $axioms(H)$ is a set of axiom instances that are used for constructing H, and $literals(H)$ is a set of literals (equality assumptions are not included) in H.

The first extension enables us to exploit the implicit information inferred by abductive inference for coreference resolution. For example, in order to realize the example inference in Figure 6.1, we can exploit a binary feature that indicates whether $eat(x)$ and $eat(y)$ are abductively inferred from the observed text or not. The second extension allows us to give a penalty for deriving "not necessarily true" information with backward inference, according to the reliability of axioms. One can give a confidence value estimated by a certain knowledge acquisition technology as the feature value. In our experiment, we use a binary feature for indicating whether an inference rule is used or not, as described in Sec. 6.4.1.

6.3 Related Work

6.3.1 Coreference Resolution

In the past decades, a lot of effort in computational linguistics and NLP was put into coreference resolution, see [Ng, 2010] for a detailed survey. Coreference resolution may require deep understanding of text, access to world knowledge, and inference ability. For example, Levesque [2011] considers twin sentences such as *Ed shouted at Tim because he crashed the car* and *Ed shouted at Tim because he was angry*. In order to resolve coreference in these sentences one requires world knowledge about people shouting when being angry and people shouting at someone who made a mistake, e.g., crashed a car.

Surprisingly, most of the contemporary coreference resolution systems, including the *Stanford NLP* system [Lee et al., 2011], the winner of CoNLL-2011

shared task: "Modeling Unrestricted Coreference in OntoNotes" [Pradhan et al., 2011], are rule-based resolvers. They encode traditional linguistic constraints on coreference and do not exploit any world knowledge. There exist attempts to resolve coreference based on world knowledge resources such as WordNet hierarchy, Wikipedia, semantic similarity, narrative chains [Irwin et al., 2011; Ng, 2007; Ponzetto and Strube, 2006; Rahman and Ng, 2012]. Unfortunately, the corresponding resolvers were either not evaluated in large-scale challenges, such as CoNLL shared task, or did not show convincing performance in the challenges. Thus, the question remains open whether employing world knowledge can improve coreference resolution in large unfiltered corpora.

6.3.2 Overmerging in Inference-based Discourse Processing

If abduction is applied to discourse processing, coreference links naturally follow as a by-product of constructing best explanations. In weighted abduction, coreference resolution is equal to unification of predications; see [Hobbs et al., 1993] or Chapter 2. Similarly, if deductive model building is applied to discourse interpretation, coreference links result from the model minimality. Both inference approaches are based on the idea that predications having the same names refer to the same entity and therefore their arguments can be set to be equal if it does not imply logical contradictions. However, in the situations when necessary knowledge is missing from the knowledge base, both the deductive and the abductive procedures are likely to miss relevant coreference links and establish wrong links (overmerge entities).

For example, given $O = animal(e_1, x) \wedge animal(e_2, y)$, weighted abduction incorrectly assumes x equals y even when $dog(e_3, x)$ and $cat(e_4, y)$ are observed. For *John runs and Bill runs*, with the observations $O = John(e_1, x) \wedge run(e_2, x) \wedge Bill(e_3, y) \wedge run(e_4, y)$, weighted abduction assumes John and Bill are the same individual just because they are both running. If we had complete knowledge about disjointness, the overmerging problem might not occur because of logical contradictions. However, it is not plausible to assume that we would have an exhaustive knowledge base.

The *overmerging* problem is a serious obstacle in applying reasoning to discourse processing, because it leads to a large number of incorrect inferences, see [Ovchinnikova, 2012] for examples. There have been attempts to employ semantic similarity for merging predications in a deductive framework [Dellert, 2011] and attempts to use linguistically motivated constraints in order to prohibit incorrect unification in an abductive framework [Ovchinnikova, 2012; Ovchinnikova et al., 2011]. However, the issue of overmerging was never systematically studied and the proposed solutions were never evaluated. In terms of this respect, our proposal can be regarded as the framework that can prohibit incorrect unification through the cost of equalities.

6.4 Evaluation

We evaluate coreference resolution in our weighted abduction framework using the CoNLL-2011 shared task dataset [Pradhan et al., 2011]. The CoNLL-2011 dataset was based on the English portion of the OntoNotes 4.0 data [Hovy et al., 2006]. OntoNotes is a corpus of large scale annotation of multiple levels of the shallow semantic structure in text. The OntoNotes coreference annotation captures general anaphoric coreference that covers entities and events not limited to noun phrases or a limited set of entity types.

The CoNLL-2011 shared task was to automatically identify mentions of entities and events in text and to link the coreferring mentions together to form entity/event chains. In our experiment, we do not identify mentions, but only compute precision and recall of the inferred coreferences links given the mentions identified in the gold standard annotation.

In the CoNLL-2011 shared task, four metrics were used for evaluating coreference performance: MUC, B^3, CEAF, and BLANC. The evaluation metrics are described in [Pradhan et al., 2011]. Each of the metric tries to address the shortcomings of the earlier metrics. MUC is the oldest metric; it has been criticized for not penalizing overmerging [Recasens and Hovy, 2010]. Since one of the goals of this study is to reduce overmerging in our inference-based framework, this metric does not seem to be representative for us. The B^3 and CEAF metrics were also considered to produce counter-intuitive results [Luo, 2005; Recasens and Hovy,

2010]. BLANC, as the most recent evaluation metric, overcomes the drawbacks of MUC, B^3, and CEAF. The definition formula of BLANC given in [Recasens and Hovy, 2010] is replicated in Table 6.1, where rn, wc, rn, wn indicate the number of right coreference links, wrong coreference links, right non-coreference links, and wrong non-coreference links correspondingly.

Score	Coreference	Non-coreference	Metric
P	$P_c = \frac{rc}{rc+wc}$	$P_n = \frac{rn}{rn+wn}$	BLANC-P $= \frac{P_c+P_n}{2}$
R	$R_c = \frac{rc}{rc+wn}$	$R_n = \frac{rn}{rn+wc}$	BLANC-R $= \frac{R_c+R_n}{2}$
F	$F_c = \frac{2P_cR_c}{P_c+R_c}$	$F_n = \frac{2P_nR_n}{P_n+R_n}$	BLANC $= \frac{F_c+F_n}{2}$

Table 6.1: Definition formula for BLANC.

We rely on BLANC when drawing conclusions, but present values of other three evaluation metrics as well.

6.4.1 Features

We derive features for resolving coreference from different knowledge sources, which are described in this section. Each feature is defined for pairs of unifiable variables (v_1, v_2). The features are summarized in Table 6.1.

Incompatible properties If two entities have incompatible properties, they are unlikely to be identical. We use WordNet antonymy (*black – white*) and sibling relation (*cat – dog*) to derive incompatible properties. Moreover, we assume that two proper names not belonging to the same WordNet synset are unlikely to refer to the same entity. Correspondingly, we generate three binary features A, S, and P (see Table 6.1).

Conditional unification If two entities have very frequent common properties, these properties usually do not represent a good evidence for the entities to be identical. For example, given *John goes* and *he goes*, it might be incorrect to assume that *John* and *he* are coreferential just because they are both going. We want to allow unification of frequent predications (e.g., *go*) only if there is other

Feature type	**Feature**
Incompatible properties	$A(v_1, v_2) = \begin{cases} 1 \text{ if } \exists p_1(.., v_1, ..), p_2(.., v_2, ..)\text{: } p_1, p_2 \text{ are} \\ \text{WN antonyms;} \\ 0 \text{ otherwise} \end{cases}$ $S(v_1, v_2) = \begin{cases} 1 \text{ if } \exists p_1(.., v_1, ..), p_2(.., v_2, ..)\text{: } p_1, p_2 \text{ are} \\ \text{WN siblings;} \\ 0 \text{ otherwise} \end{cases}$ $P(v_1, v_2) = \begin{cases} 1 \text{ if } \exists p_1(e_1, v_1), p_2(e_2, v_2)\text{: } p_1, p_2 \text{ are} \\ \text{proper names,} \\ \text{not in the same WN synset;} \\ 0 \text{ otherwise} \end{cases}$
Conditional unification	$CU(v_1, v_2) = \begin{cases} 1 \text{ if } \exists p_1(v_1, x_1, .., x_n), p_2(v_2, y_1, .., y_n)\text{:} \\ p_1, p_2 \text{ are frequent predicates} \\ \text{and } \forall i \in \{1, .., n\} : s_{x_i, y_i} = 1; \\ 0 \text{ otherwise} \end{cases}$
Argument inequality	$SA(v_1, v_2) = \begin{cases} 1 \text{ if } \exists p(.., v_1, .., v_2, ..); \\ 0 \text{ otherwise} \end{cases}$ $EA(v_1, v_2) = \begin{cases} 1 \text{ if } \exists p(v_1, .., e_1, ..), p(v_2, .., e_2, ..)\text{:} \\ s_{v_1, v_2} \wedge s_{e_1, e_2} = 0; \\ 0 \text{ otherwise} \end{cases}$
Explicit non-identity	$NI(v_1, v_2) = \begin{cases} 1 \text{ if } \exists p(e, v_1, v_2)\text{: } p \text{ is} \\ \text{a non-identity predicate;} \\ 0 \text{ otherwise} \end{cases}$
Functional relations	$FR(v_1, v_2) = \begin{cases} 1 \text{ if } \exists p(e_1, v_1, x_1), p(e_2, v_2, x_2)\text{:} \\ p \text{ is a functional relation predicate} \\ \text{and } x_1 \neq x_2 \text{ and } v_1 = v_2; \\ 0 \text{ otherwise} \end{cases}$
Modality	$M(v_1, v_2) = \begin{cases} 1 \text{ if } \lvert MCPred(v_1) \cap MCPred(v_2) \rvert = \emptyset; \\ 0 \text{ otherwise} \end{cases}$
Common properties	$CP_1(v_1, v_2) = \lvert CPred(v_1, v_2) \rvert,$ $CP_2(v_1, v_2) = \sum_{p \in CPred(v_1, v_2)} Freq(p)$ $CP_3(v_1, v_2) = \sum_{p \in CPred(v_1, v_2)} WNAbst(p)$
Derivational relation	$DR(v_1, v_2) = \begin{cases} 1 \text{ if } \exists p_1(v_1, ..), p_2(v_2, ..)\text{:} \\ p_1, p_2 \text{ are derivationally related;} \\ 0 \text{ otherwise} \end{cases}$

Table 6.1: Summary of the feature set.

evidence for their arguments to be unified. In order to capture this idea, we introduce binary feature CU and compute its value as follows: If v_1 and v_2 occur as first arguments of propositions $p_1(v_1, x_1, .., x_n), p_2(v_2, y_1, .., y_n)$, such that p_1, p_2 are frequent predicates, and $\forall i \in \{1, .., n\} : s_{x_i, y_i} = 1$ (where s is an ILP variable,

see Sec. 3) then $CU(v_1, v_2) = 1$; otherwise $CU(v_1, v_2) = 0$.

Argument inequality We use two argument constraints to generate features. First, we assume that arguments of the same proposition usually cannot refer to the same entity. Reflexive verbs represent an exception (e.g., *John cut himself*), but we assume that these cases are resolved by the *Boxer* semantic parser (see Sec. 2.4) and do not require inference. We create binary feature SA and compute its value as follows: If v_1 and v_2 occur as arguments of the same proposition then $SA(v_1, v_2) = 1$; otherwise $SA(v_1, v_2) = 0$.

One more feature we introduce concerns event variables. For example, given the sentences *John said that Mary was reading* and *John said that he was tired* we do not want to unify both *say* propositions, because in both cases something else has been said. Predicates like *say* usually have clauses as their arguments. Unifying clauses just because they are arguments of the same predicate is often incorrect. In our framework, a clause is represented by an event variable, i.e. a variable, which is a first arguments of the head of the clause. We make the following assumption: If two unifiable propositions $p(v_1, .., e_1, ..), p(v_2, .., e_2, ..)$ have event variables as their arguments, then they are unlikely to be unified if the event arguments have not been unified. We create binary feature EA and compute its value as follows: if (i) there are two unifiable propositions $p(v_1, .., e_1, ..), p(v_2, .., e_2, ..)$ that have event variables e_1, e_2 as non-first arguments, (ii) $e_1 \neq e_2$, and (iii) $v_1 = v_2$, then $EA(v_1, v_2) = 1$; otherwise $EA(v_1, v_2) = 0$.

Explicit non-identity We manually collected a set of 33 predicates indicating explicit non-identity, e.g., *similar to*, *different from*, *equal to*. Presence of these predicates in a logical form indicates that their second and third arguments are unlikely to refer to the same entity. We create binary feature NI and compute its value as follows: If there is $p(e, v_1, v_2)$ and p is a predicate indicating explicit non-identity then $NI(v_1, v_2) = 1$; otherwise $NI(v_1, v_2) = 0$.

Functional relations A binary relation r is functional if $\forall x, y_1, y_2 : r(x, y_1) \wedge r(x, y_2) \rightarrow y_1 = y_2$. For example, a person can be a son of exactly one person. Lin et al. [2010] automatically learn functional relations from a corpus and assign a

confidence score to each extracted relation. We use the set of functional relations generated by [Lin et al., 2010] in order to generate feature FR. We extract 1,661 functional relations from the dataset. We create binary feature FR and compute its value as follows: if (i) there are two predicates $p(e_1, v_1, x_1), p(e_2, v_2, x_2)$, where p indicates a functional relation, (ii) $x_1 \neq x_2$, and (iii) $v_1 = v_2$ then $FR(v_1, v_2) = 1$; otherwise $FR(v_1, v_2) = 0$.

Modality We assume that two predications having different modality are unlikely to refer to the same entity. For example, given *John runs* and *he does not/might run*, *John* and *he* are unlikely to be coreferential. Let $MPred(v)$ be a set of predicates that represent the modality of event v. In our experiments, we consider three modality-denoting predicates produced by the *Boxer* semantic parser (*nec*, *pos*, *not*), and verbal predicates (e.g., *think*) as modality-denoting predicates. We create binary feature M and compute its value as follows: if there are two unifiable verbal propositions $p(v_1, ...), p(v_2, ...)$ and $|MPred(v_1) \cap MPred(v_2)| = \emptyset$ then $M(v_1, v_2) = 1$; otherwise $M(v_1, v_2) = 0$.

Common properties We assume that the more properties two entities share the more likely it is that they are identical. For example, given *John was jogging, while Bill was sleeping. He jogs every day*, *John* and *he* are likely to be coreferential, because they are both arguments of *jog*. Let $CPred(v_1, v_2)$ be a set of pairs of predicates p_1, p_2, such that v_1, v_2 occur at the same argument positions of p_1 and p_2 while p_1 are p_2 equal or they occurs in the same WordNet synset. We generate three types of real-valued features: $CP_1(v_1, v_2) = |CPred(v_1, v_2)|$, $CP_2(v_1, v_2) = \sum_{p \in CPred(v_1, v_2)} Freq(p)$, and $CP_3(v_1, v_2) = \sum_{p \in CPred(v_1, v_2)} WNAbst(p)$, where $Freq(p)$ is a word-frequency of p from the Corpus of Contemporary American English[1], and $WNAbst(p)$ is a level of abstraction of p in the WordNet hierarchy (the number of steps to the root).

Derivational relations We use WordNet derivational relations between nouns and verbs in order to link nominalizations and verbs. For example, given *Sales of*

[1] http://www.wordfrequency.info/

cars grew. The growth followed year-to-year increases, *grew* and *growth* are coreferential. We generate binary feature DR to capture these links (see Table 6.1).

6.4.2 Knowledge for Inference

Abductive reasoning procedure is based on a knowledge base consisting of a set of axioms. In the experiment described in this chapter we employed following background knowledge.

WordNet The dataset we use for evaluation (see Sec. 6.4) is annotated with WordNet [Fellbaum, 1998] senses. Given this annotation, we mapped word senses to WordNet synsets. Given WordNet relations defined on synsets, we generate axioms of the following form:

Hyperonymy, instantiation: $synset_1(s_1, x) \rightarrow synset_2(s_2, x)$
Causation, entailment: $synset_1(s_1, e_1) \rightarrow synset_2(s_2, e_2)$
Meronymy, membership: $synset_1(s_1, x_1) \rightarrow synset_2(s_2, x_2) \wedge of(x_1, x_2)$

We extract 22,815 axioms from WordNet.

FrameNet We generated axioms mapping predicates with their arguments into FrameNet [Ruppenhofer et al., 2010] frames and roles. For example, the following axiom maps the verb *give* to the GIVING frame.

$\text{GIVING}(e_1) \wedge \text{DONOR}(e_1, x_1) \wedge \text{RECIPIENT}(e_1, x_2) \wedge \text{THEME}(e_1, x_3) \rightarrow give(e_1, x_1, x_3) \wedge to(e_2, e_1, x_2)$

Weights of these axioms are based on frequencies of lexeme-frame mappings in the annotated corpora provided by the FrameNet project. Moreover, we used FrameNet frame relations to derive axioms. An example of an axiomatized relation is given below.

$\text{GIVING}(e_1) \wedge \text{DONOR}(e_1, x_1) \wedge \text{RECIPIENT}(e_1, x_2) \wedge \text{THEME}(e_1, x_3) \quad \rightarrow$
$\text{GETTING}(e_2) \wedge \text{SOURCE}(e_2, x_1) \wedge \text{RECIPIENT}(e_1, x_2) \wedge \text{THEME}(e_1, x_3)$

In order to generate the FrameNet axioms, we used the previous work on axiomatizing FrameNet [Ovchinnikova, 2012]. We generated 12,060 axioms from the dataset. In addition, we used a resource assigning possible lexical fillers disambiguated into WordNet synsets to FrameNet roles [Bryl et al., 2012]. For

example, the role THEME of the GIVING frame is mapped to synsets `object#n#1` and *thing#n#1*. Given this information, the following axiom is generated.

$thing\#n\#1(s, x) \rightarrow \text{GIVING}(e_1) \wedge \text{THEME}(e_1, x)$

Weights of these axioms are based on the scores provided by [Bryl et al., 2012]. We generated 24,571 axioms from the dataset.

Narrative chains Similar to [Rahman and Ng, 2012], we employ narrative chains learned by Chambers and Jurafsky [2009], which were shown to have impact on resolving complex coreference; see [Rahman and Ng, 2012] for details. Narrative chains are partially ordered sets of events centered around a common protagonist that are likely to happen in a sequence. Knowledge about such sequences can facilitate coreference resolution. For example, given *Max fell, because John pushed him* we know that *Max* and *him* are coreferential, because we know that an object of the pushing event can be a subject of the falling event. For example, we generate the following axioms.

$Script\#1(s, e_1, x_1, u) \rightarrow \text{arrest}(e_1, x_1, x_2, x_3) \wedge police(e_2, x_1)$

$Script\#1(s, e_1, x_1, u) \rightarrow \text{charge}(e_1, x_1, x_2, x_3) \wedge police(e_2, x_1)$

Weights of these axioms are based on the scores provided by Chambers and Jurafsky [2009]. We extract 1,391,540 axioms from the dataset.

6.4.3 Disambiguation of Named Entities

In the experiment on coreference resolution, we extended *Boxer*'s output with the information inferred by the *AIDA* tool. The *AIDA* tool [Yosef et al., 2011] is a framework for entity detection and disambiguation. Given a natural language text, it maps mentions of ambiguous names onto canonical entities like people or places, registered in a knowledge base like DBpedia [Bizer et al., 2009] or YAGO [Suchanek et al., 2008]. For example, mentions *A. Einstein* and *Einstein* will be both mapped to the YAGO node *Albert_Einstein*. An add-on to our pipeline assigns the same variables to each two named entities disambiguated by *AIDA* into the same YAGO node.

6.4.4 Results and Discussions

We intend to evaluate whether introduction of linguistically motivated features (Sec. 6.4.1) and world knowledge (Sec. 6.4.2) enables us to outperform the naive inference-based approach implying that predications with the same names refer to the same entities. In order to evaluate the impact of each feature and knowledge component separately, we run ablation tests.

Note that for 145 of 6,894 sentences in the test set, no logical forms were produces by the *Boxer* semantic parser. Moreover, in the run employing WordNet-based inference, inference results could not be produced for 101 of 303 test texts because of the computational complexity of reasoning. In order to keep the comparison fair, we use evaluate all features and knowledge components on the same set of 202 texts, for which inference results were produced in all runs.

Table 6.2 represents the results of the ablation tests. We test the features listed in Table 6.1 as well as axioms extracted from WordNet (WN), FrameNet (FN), narrative chains (NC) and knowledge provided by AIDA (AI). All features representing incompatible properties are tested together (IP in Table 6.2). Similarly, all argument inequality features (AI) and common property features (CP) are tested together.

The first row in the table represents results for to the run without employing any features and knowledge resources. In the second run, world knowledge is employed without linguistic features. These two runs correspond to the original weighted abduction approach to unification implying unification of all predications having the same predicate names. We see that adding knowledge does not result in higher values of BLANC. This happens because of the overmerging problem increased by additional coreference links inferred with the help of the employed knowledge resources.

Then we test linguistic features intended to block incorrect unification (IP, CU, AI, NI, FR, M) one by one. Each of the features improves the BLANC values; conditional unification CU has the most significant impact. The common property feature (CP) and the derivational relations feature (M) introduce additional unifications. Therefore we test them together with the best combination of the unification blocking features (IP+CU+AI+NI+FR+M). Both features

have a positive impact as compared to the run employing just the unification blocking features. Now we test each world knowledge component using the best combination of features (IP+CU+AI+NI+FR+M+CP). Again, each knowledge component has a positive impact in terms of BLANC as compared to the run using the best combination of all features.

Features								Inference				BLANC		
IP	*CU*	*AI*	*NI*	*FR*	*M*	*CP*	*DR*	WN	FN	NC	AI	R	P	F
												53.0	51.7	39.1
								√	√	√	√	52.3	51.3	39.9
√												53.5	51.9	41.0
	√											55.7	60.9	56.6
		√										53.0	51.6	41.0
			√									52.8	51.5	40.5
				√								52.9	51.6	40.7
					√							53.3	51.7	41.0
√	√	√	√	√	√	√						58.4	61.6	59.4
√	√	√	√	√	√		√					57.5	61.4	58.6
√	√	√	√	√	√	√	√	√				57.4	61.2	58.5
√	√	√	√	√	√	√	√		√			59.5	61.0	60.1
√	√	√	√	√	√	√	√			√		59.0	61.5	59.9
√	√	√	√	√	√	√	√				√	59.7	61.5	60.4
√	√	√	√	√	√	√	√	√	√	√	√	59.9	60.9	60.3

Table 6.2: Ablation tests of features and world knowledge.

The results of the ablation tests show significant improvement over the naive approach (by more than 20% F-measure), but can we claim that we solved the overmerging problem? We perform one more experiment in order to get a deeper understanding of the performance of our discourse processing pipeline in coreference resolution.

As already mentioned, the best performance in the CoNLL-2011 shared task was achieved by the *Stanford NLP* system [Lee et al., 2011]. We replicate the results of *Stanford NLP* as applied to the CoNLL-2011 dataset; see the first row in Table 6.3. We use the output of *Stanford NLP* only for those texts, which could be processed by our discourse processing pipeline, therefore the recall/precision values for *Stanford NLP* in Table 6.3 are lower than the original results published in [Lee et al., 2011].

We aim at checking whether enriching the output of the state-of-the-art coreference resolver with additional links inferred by our system using all features and all world knowledge will improve the performance. The evaluation of the "merged" output is presented in the second row of Table 6.3 (SNLP+WA). Un-

fortunately, we have to admit that the precision of SNLP+WA is lower than that of SNLP alone. This happens because adding world knowledge results in new coreference links, but the overmerging problem is not completely solved. SNLP discovers 2277 out of 7557 correct coreference links and 40247 out of 41527 correct non-coreference links. In the merged output, there are more correct coreference links (3065), but less correct non-coreference links (36959).

System	MUC			B^3			CEAFE			BLANC		
	R	P	F	R	P	F	R	P	F	R	P	F
SNLP	42.8	74.4	54.3	50.4	85.2	63.4	66.3	32.6	43.7	63.5	76.2	66.7
SNLP+WA	52.0	70.1	59.7	57.3	72.7	64.1	60.5	37.2	46.1	64.8	64.7	64.7

Table 6.3: Performance of the *Standford NLP* system (SNLP) compared to performance of our weighted abduction engine enriched with *Standford NLP* (SNLP+WA) output.

The main cause of overmerging is related incompatible properties. We anticipated the incompatible properties to have a more significant impact on precision than they actually had in the ablation tests. But in the current study, we consider only those properties to be incompatible, which are expressed syntactically in the same way, e.g., *Japanese goods* vs. *German goods*. However, the same property can be expressed by a wide variety of syntactic constructions, e.g., *goods from Germany*, *goods produced in Germany*, *Germany produced goods* etc. In order to discover deeper contradictions, we have to work on normalization of the representation of properties, e.g., use $origin{:}Germany{:}x$ instead of $German(e, x)$ and $from(e_1, x, y) \wedge Germany(e_2, y)$. FrameNet attempts to achieve such a normalization by using standardized frame and role names. Unfortunately, the limited coverage of the FrameNet resource [Cao et al., 2008; Shen and Lapata, 2007] does not allow us to solve the problem on a large scale.

Analyzing the results, we also found overmergings not implying any explicit contradictions. For example, in the sentence *He sat near him*, both *he* prepositions are unlikely be coreferential, but our framework fails to capture it. Such overmergings might be blocked by explicit modeling of discourse salience. In the future, we plan to use existing discourse salience models (e.g., [Lappin and Leass, 1994]) to create real-valued salience features for weighted unification.

One more issue concerns the quality of the obtained interpretations. Our learning framework assumes that we can obtain optimal solutions, but we also

exploit suboptimal solutions by imposing a timeout in this experiment. However, it has been reported that exploiting suboptimal solutions sometimes hurts performance [Finley and Joachims, 2008]. In the future, we will address this problem using an approximate learning framework (e.g., [Huang et al., 2012]).

6.5 Conclusions

In this chapter, we explored an inference-based coreference resolution model. In our framework, resolving coreference is a by-product of constructing best interpretations of text. Traditional approaches to coreference resolution have exploited world knowledge to capture semantic compatibility between mentions, encoding them as a feature vector for machine learning-based classifiers. However, there are many cases where there exist many cases where several antecedents are syntactically or semantically compatible with an anaphoric expression.

In this chapter, to remedy this problem, we have proposed a machine learning-based hybrid model that combines the traditional compatibility-based approach with a logical inference-based approach. We recast the problem of implicit event derivation and coreference resolution as the problem of abductive explanation finding, integrating those two approaches to complement the weakness of each approach with an abductive inference framework. Our empirical evaluation shows that the implicit event information improves the performance of compatibility feature-based coreference resolution model.

However, the use of implicit event information is not as effective as we expected. Our additional investigation revealed that explanations are certainly generated for each problem, but we observed that the explanations generated by the current knowledge base are not really useful for coreference resolution. Our future direction includes a direct evaluation of the quality of implicit event information inferred by the system to figure out what kind of knowledge is still not enough for inferring useful information for coreference resolution.

Chapter 7

Conclusions

7.1 Summary

In this thesis, we have addressed one of the big issues in natural language processing research: *what mechanism enables us to use world knowledge effectively in discourse processing?* To find out the answer of this question, we have worked on the main hypothesis that *inference*-based approaches would be better alternative mechanisms to conventional feature-based approaches. As an inference framework, we have focused on a particular mode of inference, namely *abduction*.

The key contribution of this thesis can be summarized as follows:

(i) we propose an efficient inference method of first-order logic-based abduction that avoids computationally expensive grounding procedures, showing how to directly formulate the abductive inference problem on first-order logic as an Integer Linear Programming (ILP) optimization problem;

(ii) we show how to formulate the machine learning problem of first-order logic-based abduction with the framework of online large-margin training, which has been shown to have both predictive performance and scalability to larger problems;

(iii) we propose a novel hybrid model that combines the conventional feature-based approach with a logical inference-based approach with an abductive

inference, giving a detailed comparison of feature-based approaches and inference-based approaches in both qualitative and empirical ways.

In Chapter 2, we give a basic idea of inference-based discourse processing. In particular, we elaborate on the *Interpretation as Abduction* framework [Hobbs et al., 1993], an pioneering work of *abduction*-based approach to discourse processing. There have been two big obstacles to apply abduction-based discourse processing to real-life problems: (i) how to search the best abductive explanation efficiently and (ii) how to train the score function in a supervised manner. In order to verify our main hypothesis, we first address these two problems in the next two chapters.

In Chapter 3, we have proposed an ILP-based formulation for cost-based abduction in first-order predicate logic. Although FOL reasoning is computationally expensive, the proposed optimization strategy exploits two techniques to improve the inefficiency. The first technique is *lifted inference*, where inference on first-order logic is directly performed without grounding. The second technique is *cutting plane inference*, which is an iterative optimization procedure for large constrained optimization problems. These techniques bring us to a significant boosting of the efficiency of the reasoner. We have evaluated our method on two datasets, including real-life NLP problems (i.e. RTE dataset with axioms generated from WordNet and FrameNet). Our evaluation revealed that our inference method was more efficient than the other existing abductive reasoners.

In Chapter 4, we proposed a supervised approach for training the abductive score function. We formulated the learning problem as the problem of discriminative structured learning with latent variables. More specifically, we modeled the score function as a weighted linear feature function, and then apply Passive Aggressive algorithm [Crammer et al., 2006], an online large-margin training algorithm. In our evaluation, we demonstrated that our learning procedure could reduce the loss, and improved the predictive performance of story understanding tasks in both open tests and closed tests.

Since the proposed methods in Chapter 3 and Chapter 4 could overcome the two obstacles of real-life abductive discourse processing, we conducted a detailed comparison of feature-based approaches and inference-based approaches by taking anaphora resolution as the subject of our case study in the next two chapters. In

the case study, we use the techniques developed in Chapter 3 and Chapter 4 for inference and learning.

In Chapter 5, we first proposed a feature-based anaphora resolution model and then discussed the problem of feature-based approaches. From the detailed error analysis of our model, we found out that there exist many cases where several antecedents are syntactically or semantically compatible with an anaphoric expression, and the feature-based approaches are not guaranteed to identify correct antecedents in such situations. This kind of problems is named *difficult coreference problems* in Chapter 6.

In Chapter 6, we have proposed an inference-based coreference resolution model that improves the limitations of feature-based anaphora resolution model and handles difficult coreference problems. We propose a machine learning-based hybrid model that combines the conventional feature-based approach with a logical inference-based approach. We integrate those two approaches to complement the weakness of each approach, using an abductive inference framework. In the evaluation, we found that our inference-based coreference resolution model improved the performance of coreference resolution model. However, the use of implicit event information is not as effective as we expected. We suspect that the generated explanations are still not useful enough for coreference resolution.

7.2 Future Directions

Let us go back to the main question of this thesis and try to answer the question: what mechanism enables us to exploit world knowledge resources for discourse processing in a maximally effective way. To answer this question, we have worked on the main hypothesis that inference-based approaches would be better alternatives to feature-based approaches in this thesis. To answer whether the hypothesis is proven to be true or false, we believe that the answer is "the hypothesis is partially explained (proven to be true), namely the hypothesis is still assumed with a small cost."

Why is it still "assumed"? As shown in the results of the case study in Chapter 6, the effect of using inference is not as effective as we expected. From the experiments, we can see the following three problems. The first problem is

about the insufficiency of knowledge resources. Our additional analysis after the experiments revealed that we could generate an abductive explanation for about a half of the texts, but most of them cannot be used as a useful clue for coreference resolution.

The second problem is the issue of computational efficiency of abductive reasoner. We observed that about a half of the coreference problems were not solved within 60 seconds even if the depth of backward-chaining was limited to two steps. We found that the bottleneck is the search-space generation process, which is a process of generating a set of potential elemental explanations (see Chapter 3 for detail).

The third problem is about the meaning representation of natural language texts. The current meaning representation we rely on is almost close to the surface expression. As a result, we get two different meaning representations for linguistic expressions that denote the same meaning (e.g. *Japanese goods* and *Goods produced in Japan* are converted into $goods(x) \wedge japanese(x)$ and $goods(x) \wedge produce(e, u, x) \wedge in(e, y) \wedge japan(y)$ respectively), which makes our reasoning process error prune.

In the next subsections, we elaborate on how to address these issues in future work.

7.2.1 Harvesting World Knowledge for Events

We found that there are few inference rules for event–event relations which are needed for identifying coreference in our knowledge base, such as causal relation, purpose-means relation, and presupposition relation. As a solution to the insufficiency problem of knowledge sources, we attempt to take two solutions.

The first option is to extract more inference rules from ConceptNet5.[1] ConceptNet5 is a large commonsense knowledge base, which is derived from different knowledge sources such as ReVerb,[2] WordNet, or OpenMind project. However, ConceptNet5 does *not* provide us the coreference relations between arguments in two concepts. For example, the concept *have* is related to the concept *eat*

[1] `http://conceptnet5.media.mit.edu/`
[2] `http://reverb.cs.washington.edu/`

with *MotivatedByGoal* relation, but it does not tell us the subject/object of *eat* corresponds to the subject/object of *have*. Therefore, in order to convert these relations into the logical forms, we need to estimate which arguments in a concept corresponds to arguments in another concept (e.g. $eat(X, Y) \Rightarrow have(X, Y)$ v.s. $eat(X, Y) \Rightarrow have(Y, X)$). We plan to use a distributional hypothesis-based approach, similarly to DIRT score [Lin and Pantel, 2001].

The DIRT score calculates the likelihood of inference rules, based on the extended distributional hypothesis: that is, given inference rule $X\ rel_1\ Y \leftrightarrow X\ rel_2\ Y$, the rule is plausible if a set of instantiations of each corresponding argument is similar. For example, *X solve Y* $\leftrightarrow$ *X is a solution of Y* is plausible, because the instances of the subject position of *solve* would be similar to the instances of the subject position of *is a solution of*.

The second option we consider is to use an abstract representation for verbs and then perform inference on the abstract level, using axioms defined on the abstract level, such as deep lexical semantics in [Hobbs, 2008]. This generalization would allow us to alleviate the sparsity problem of inference rules.

7.2.2 Comparing Abductive Approach with Deductive Approach for Discourse Processing

A recent study [Raghavan et al., 2012] proposes a probabilistic deductive inference approach for discourse processing. Raghavan et al. [2012] use Bayesian Logic Programs (BLPs) [Kersting and Raedt, 2001] to infer implicit information from observed texts. The key difference to an abductive inference approach is that abductive inference does *not* commit to the truth value of propositions if there is no information enough to determine the truth value of these propositions (see the discussion of specificity in Chapter 2 for more detail).

However, it is a non-trivial issue whether this property has a big impact on the quality of inferred explanations or not. It will be interesting to compare the output of explanations generated by abduction with probabilistic conclusions generated by deduction.

7.2.3 Applying Cutting Plane Inference for Search-space Generation

To address the issue of efficiency of the reasoner, we plan to apply Cutting Plane Inference to both the search-space generation and ILP inference. More specifically, we repeat the generation of potential elemental explanations and ILP optimization interactively, as in cutting plane MAP inference in MLNs [Riedel, 2008].

Currently, the cutting plane inference is applied to the ILP optimization step in the proposed method in Chapter 3. However, applying it to both the search space generation and ILP optimization makes the reasoner more efficient, because we found out that the search-space generation is the bottleneck of our approach, as mentioned earlier.

7.2.4 Normalizing Meaning Representations

In order to address the issue of meaning representation, we plan to normalize the logical forms in observations and knowledge bases. For example, we plan to make *Japanese goods* and *Goods produced in Japan*, which are currently converted into $goods(x) \wedge japanese(x)$ and $goods(x) \wedge produce(e, u, x) \wedge in(e, y) \wedge japan(y)$, to have the same logical forms such as $origin : Japan : x$. FrameNet attempts to achieve such a normalization by using standardized frame and role names. However, the limited coverage of the FrameNet resource [Cao et al., 2008; Shen and Lapata, 2007] does not allow us to solve the problem on a large scale.

7.2.5 Handling Linguistic Expressions of Logical Connectors and Quantifiers

We plan to elaborate our treatment of natural language expressions standing for logical connectors and quantifiers such as *if*, *not*, *or*, *all*, *each*, and others. Moreover, modality requires special treatment. This advance is needed in order to achieve more precise inferences, which are at the moment based on our approach to the unification of the core information content ("aboutness") of texts.

7.2.6 Evaluating Abductive Explanations

There are two options to achieve the evaluation of abductive explanations inferred by the system. The first option is an intrinsic evaluation, namely human evaluators directly check whether the generated explanations are good or not. The second option is an extrinsic evaluation. In the extrinsic evaluation, we assume to use an abductive explanation as a clue of a certain NLP task and check whether using the information of generated explanations improves the predictive performance of the NLP task or not.

In this thesis, we adopted the second option. We evaluated abductive explanations in terms of whether they provide a useful clue for coreference resolution or not. As another extrinsic evaluation, we intend to use the task of Recognizing Textual Entailment (RTE), one of the knowledge-intensive natural language processing tasks. In RTE, the system is given a text (T) and a hypothesis (H) and must decide whether the hypothesis is entailed by the text plus commonsense knowledge. Because the previous study [Hickl and Bensley, 2007] shows that inferring implicit information inferred from texts plays an important role in RTE, it would be a good test bed for evaluating abductive explanations. We also plan to evaluate our inference-based coreference resolution model on a dataset from [Rahman and Ng, 2012],[1] which is a set of difficult coreference problems.

Our future direction also includes an intrinsic evaluation of abductive explanations. However, to the best of our knowledge, there are few previous study that directly evaluates inferred implicit information. A series of studies in Machine Reading (MR) [Etzioni et al., 2006; Penas and Hovy, 2010, etc.] projects pursues implicit information extraction from natural language texts. We first plan to evaluate our system on the task of MR. We also intend to create our own corpus for evaluating inferred information. As a first step, we will evaluate generated explanations by human evaluators to see what is needed for constructing the evaluation corpus (e.g. the task formulation, a manual that achieves consistent annotation).

[1] `http://www.hlt.utdallas.edu/~vince/data/emnlp12/`

Proof of Theorem 3.3.1

Note that we do not have to consider all the ILP constraints here. As mentioned in Sec. 3.2, our constraints consist of three types: (i) for ensuring that H_S explains observation (Constraint 1), (ii) for ensuring that H_S is consistent (Constraint 2, 7), and (iii) for implementing a cost function in weighted abduction (the rest); as the reader can see, the proof of soundness and completeness is related to only (i) and (ii).

Let α_1 be a proposition that H_S satisfies $H_S \cup B \models O$, and α_2 be a proposition that H_S satisfies $H_S \cup B \not\models \bot$. Let β_i be a proposition that S_H satisfies Constraint i described in Sec. 3.2 and Sec. 3.3.1. For convenience, we repeat the constraints to be mentioned below:

- Constraint 1: $(\forall p \in O) h_p = 1$
- Constraint 2: $(\forall x, y, z \in T) h_{x=y} = 1 \wedge h_{y=z} = 1 \Rightarrow h_{x=z} = 1$
- Constraint 7: $h_{q(x_1,x_2,...,x_n)} = 1 \wedge h_{\neg q(y_1,y_2,...,y_n)} = 1 \Rightarrow (\exists i \in \{1, 2, ...n\}) h_{x_i=y_i} = 0$

For readability, we translated the ILP constraints into equivalent logical constraints. The proof of equivalence between ILP constraints and logical formulae can be found in Santos [1994]. Using these notations, the proposition that we need to prove can be expressed as $\alpha_1 \wedge \alpha_2 \Leftrightarrow \beta_1 \wedge \beta_2 \wedge \beta_7$.

(i) $\alpha_1 \wedge \alpha_2 \Rightarrow \beta_1 \wedge \beta_2 \wedge \beta_7$: it is clear that all the ILP constraints above are not violated given that H_S explains O and is consistent.

(ii) $\beta_1 \wedge \beta_2 \wedge \beta_7 \Rightarrow \alpha_1 \wedge \alpha_2$: we prove that $\beta_1 \wedge \beta_2 \wedge \beta_7 \wedge (\neg\alpha_1 \vee \neg\alpha_2)$ leads to contradiction in the following.

First, we consider the case of $\neg\alpha_1$. This implies that $H_S \cup B \not\models O$. However, by β_1, we conclude $H_S \cup B \models O$. Therefore, the $\beta_1 \wedge \beta_2 \wedge \beta_7$ and $\neg\alpha_1$ cannot be

true at the same time.

Second, we consider the case of $\neg\alpha_2$, namely $H_S \cup B \models \bot$. By the definition of *inconsistency* in propositional logic theory, this implies $(H_S \cup B \models \phi) \wedge (H_S \cup B \models \neg\phi)$, where ϕ is a logical formula. We have two cases that let this formula true. The first case is that two contradictory literals or equalities cause inconsistency: there exists the atom A such that $H_S \cup B \models A \wedge \neg A$, or the pair x, y of logical atomic terms such that $H_S \cup B \models x = y \wedge x \neq y$. However, by β_7, for any atoms A, its positive literal A and negative literal $\neg A$ cannot be hypothesized simultaneously in S_H. Also, for the pair x, y of logical atomic terms, $x = y$ and $x \neq z$ cannot be hypothesized simultaneously. The second case is the violation of equality axioms:[1] $\exists(x, y, z \in T)[H_S \cup B \models (x = y \wedge y = z)] \wedge (H_S \cup B \not\models x = z)]$. However, by β_2, for all x, y, z in S_H must satisfy transitivity, namely $\forall(x, y, z)[H_S \cup B \models (x = y \wedge y = z)] \Rightarrow (H_S \cup B \models x = z)$. Since both cases cannot be true, $\beta_1 \wedge \beta_2 \wedge \beta_7$ and $\neg\alpha_2$ cannot be true at the same time.

Since neither case can be true, we therefore conclude that H_S is a candidate explanation if and only if S_H satisfies the ILP constraints 1, 2, and 7. $\square$

[1]We omit the proof of reflexivity, symmetricalness because it is trivial by the definition of the ILP variable s.

References

A. M. Abdelbar and M. Hefny. An efficient lp-based admissible heuristic for cost-based abduction. *JETAI*, 17(3):297–303, 2005. 43

K. Apt and M. van Emden. Contributions to the Theory of Logic Programming. *Journal of the Association for Computing Machinery*, 29(3):841–862, July 1982. 32

F. B. Baldwin. *Cogniac: a discourse processing engine*. PhD thesis, 1995. 68

C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. Dbpedia - a crystallization point for the web of data. *J. Web Sem.*, 7 (3):154–165, 2009. 122

J. Blythe, J. R. Hobbs, P. Domingos, R. J. Kate, and R. J. Mooney. Implementing Weighted Abduction in Markov Logic. In *IWCS*, pages 55–64, 2011. 6, 38, 41, 45, 47, 60, 61

J. Bos. Wide-Coverage Semantic Analysis with Boxer. In J. Bos and R. Delmonte, editors, *Proceedings of STEP*, Research in Computational Semantics, pages 277–286. College Publications, 2008. 15, 39, 59

S. E. Brennan, M. W. Friedman, and C. J. Pollard. A centering approach to pronouns. In *Proceedings of the 25th annual meeting on Association for Computational Linguistics*, pages 155–162, 1987. 68

V. Bryl, S. Tonelli, C. Giuliano, and L. Serafini. A novel framenet-based resource for the semantic web. In *SAC*, pages 360–365, 2012. 121, 122

R. Bunescu. Associative anaphora resolution: A web-based approach. In *Proceedings of the EACL 2003 Workshop on The Computational Treatment of Anaphora*, pages 47–52, 2003. 69

D. De Cao, D. Croce, M. Pennacchiotti, and R. Basili. Combining word sense and usage for modeling frame semantics. In *Proc. of STEP 2008*, pages 85–101, 2008. 125, 132

N. Chambers and D. Jurafsky. Unsupervised Learning of Narrative Schemas and their Participants. In *ACL*, pages 602–610, 2009. 122

E. Charniak and R. P. Goldman. A Probabilistic Model of Plan Recognition. In *AAAI*, pages 160–165, 1991. 5, 14, 16, 21, 109

C. Cherry and C. Quirk. Discriminative, syntactic language modeling through latent svms. In *In AMTA*, 2008. 53, 62

S. T. Chivers, G. A. Tagliarini, and A. M. Abdelbar. An Evolutionary Optimization Approach to Cost-Based Abduction, with Comparison to PSO. In *IJCNN*, pages 2926–2930, 2007. 5, 21, 43

T. Chklovski and P. Pantel. VerbOcean: Mining the Web for Fine-Grained Semantic Verb Relations. In D. Lin and D. Wu, editors, *Proceedings of EMNLP*, pages 33–40. ACL, 2004. 2

H. H. Clark. Bridging. In *Thinking: Readings in Cognitive Science*, 1977. 64, 69

K. Crammer, O. Dekel, J. Keshet, and Y. Singer S. Shalev-Shwartz. Online Passive-Aggressive Algorithms. pages 551–585, 2006. 6, 9, 48, 52, 55, 128

G. B. Dantzig, R. Fulkerson, and S. M. Johnson. Solution of a large-scale traveling salesman problem. *Operations Research*, 2(4):393–410, 1954. 36

D. Davidson. The Logical Form of Action Sentences. In N. Rescher, editor, *The Logic of Decision and Action*, pages 81–120. University of Pittsburgh Press, 1967. 15

J. Dellert. Challenges of Model Generation for Natural Language Processing. Master's thesis, University of Tübingen, 2011. 116

P. Denis and J. Baldridge. Specialized models and ranking for coreference resolution. In *Proceedings of Empirical Methods in Natural Language Processing*, pages 660–669, Honolulu, Hawaii, October 2008. Association for Computational Linguistics. 77

A. Doncescu, K. Inoue, and T. Sato. Hypothesis-Finding in Systems Biology. *ALP Newsletter*, 21:2–3, 2008. 46

O. Etzioni, M. Banko, and M. J. Cafarella. Machine reading. In *Proceedings of AAAI*, 2006. 133

C. Fellbaum. WordNet: an electronic lexical database. 1998. 2, 39, 59, 69, 121

P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Trans.*, 32 (9):1627–1645, 2010. 53, 62

T. Finley and T. Joachims. Supervised clustering with support vector machines. In *ICML*, pages 217–224, 2005. 60, 113

T. Finley and T. Joachims. Training structural svms when exact inference is intractable. In *Proceedings of the 25th international conference on Machine learning*, ICML '08, pages 304–311, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-205-4. 126

N. Ge, J. Hale, and E. Charniak. A statistical approach to anaphora resolution. In *Proceedings of the 6th Workshop on Very Large Corpora*, pages 161–170, 1998. 68

B. J. Grosz, S. Weinstein, and A. K. Joshi. Centering: a framework for modeling the local coherence of discourse. *Computational Linguistics*, 21(2):203–226, 1995. 69

C. Guinn, W. Shipman, and E. Addison. The Parallelization of Membrane Computers to Find Near Optimal Solutions to Cost-Based Abduction. In *GEM*, pages 241–2–47, 2008. 43

J. A. Hawkins. *Definiteness and Indefiniteness: A Study in Reference and Grammaticality Prediction*. Croom Helm Linguistic Series. Taylor & Francis, 1978. 71

A. Hickl and J. Bensley. A Discourse Commitment-Based Framework for Recognizing Textual Entailment. In *Proceedings of ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 171–176, 2007. 133

J. R. Hobbs. Deep lexical semantics. In *Proceedings of the 9th international conference on Computational linguistics and intelligent text processing*, pages 183–193, 2008. 131

J. R. Hobbs. Ontological promiscuity. In *ACL*, pages 61–69, Chicago, Illinois, 1985. 15, 39

J. R. Hobbs, M. Stickel, P. Martin, and D. Edwards. Interpretation as abduction. *Artificial Intelligence*, 63:69–142, 1993. 3, 8, 11, 14, 16, 17, 18, 20, 39, 59, 109, 110, 113, 115, 128

T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the Twenty Second Annual International SIGIR Conference on Research and Development in Information Retrieval*, pages 50–57, 1999. 80

D. Hovy, C. Zhang, E. Hovy, and A. Penas. Unsupervised discovery of domain-specific knowledge from text. In *ACL*, pages 1466–1475, 2011. 2

E. Hovy, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel. Ontonotes: The 90% solution. In *HLT-NAACL 2006*, pages 57–60, 2006. 116

L. Huang, S. Fayong, and Y. Guo. Structured perceptron with inexact search. In *HLT-NAACL*, pages 142–151, 2012. 126

T. N. Huynh and R. J. Mooney. Max-Margin Weight Learning for Markov Logic Networks. In *SRL*, 2009. 6, 47

R. Iida, K. Inui, Y. Matsumoto, and S. Sekine. Noun phrase coreference resolution in japanese based on most likely candidate antecedents noun phrase coreference resolution in japanese based on most likely antecedent candidates. In *Journal of Information Processing Society of Japan*, pages 831–844, 2005. 64, 68, 75, 77

R. Iida, M. Komachi, K. Inui, and Y. Matsumoto. Annotating a japanese text corpus with predicate-argument and coreference relations. In *Proceedings of the ACL 2007 Workshop on Linguistic Annotation Workshop*, pages 132–139, 2007. 98

K. Inoue. Induction as consequence finding. *Machine Learning*, 55(2):109–135, 2004. 43

N. Inoue and K. Inui. ILP-Based Reasoning for Weighted Abduction. In *Proceedings of AAAI Workshop on Plan, Activity and Intent Recognition*, 2011. 109

N. Inoue and K. Inui. Large-scale Cost-based Abduction in Full-fledged First-order Predicate Logic with Cutting Plane Inference, 2012. 48, 61

J. Irwin, M. Komachi, and Y. Matsumoto. Narrative schema as world knowledge for coreference resolution. In *the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 86–92, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. 115

H. Isahara, F. Bond, K. Uchimoto, M. Utiyama, and K. Kanzaki. Development of the japanese wordnet. In *Proceedings of the 6th International Language Resources and Evaluation*, 2008. 77

M. Ishizuka and Y. Matsuo. SL Method for Computing a Near-optimal Solution using Linear and Non-linear Programming in Cost-based Hypothetical Reasoning. In *PRCAI*, pages 611–625, 1998. 5, 21, 43

T. Aviv J. Berant and J. Goldberger. Global Learning of Typed Entailment Rules. In *ACL*, pages 610–619, 2008. 36

A. Kakas, R. Kowalski, and F. Toni. Abductive logic programming. *Journal of logic and computation*, 2(6):719–770, 1992. 43

H. Kamp and U. Reyle. *From Discourse to Logic: Introduction to Model-theoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Studies in Linguistics and Philosophy. Kluwer, Dordrecht, 1993. 15

R. J. Kate and R. J. Mooney. Probabilistic abduction using markov logic networks. In *Proceedings of the IJCAI-09 Workshop on Plan, Activity, and Intent Recognition (PAIR-09)*, Pasadena, CA, July 2009. 6, 38, 41, 45, 47, 56, 60, 61

K. Kersting and L. De Raedt. Bayesian logic programs. Technical report, 2001. 6, 47, 131

R. Kowalski. Predicate Logic as a Programming Language. In *IFIP Congress*, pages 569–574, 1974. 25

T. Kudo and H. Kazawa. *Web Japanese N-gram Version 1*. Gengo Shigen Kyokai, 2007. 80

S. Lappin and H. J. Leass. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4):535–561, 1994. 125

C. Lee. *A completeness theorem and a computer program for finding theorems derivable from given axioms*. PhD thesis, 1967.

H. Lee, Y. Peirsman, A. Chang, N. Chambers, M. Surdeanu, and D. Jurafsky. Stanford's multi-pass sieve coreference resolution system at the CoNLL-2011 shared task. In *CoNLL*, pages 28–34. Association for Computational Linguistics, 2011. 114, 124

H. J. Levesque. The Winograd Schema Challenge. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*, 2011. 114

D. Lin and P. Pantel. Dirt: discovery of inference rules from text. In *KDD '01: Proceedings of the seventh ACM SIGKDD international conference*, pages 323–328, 2001. 131

T. Lin, M., and O. Etzioni. Identifying Functional Relations in Web Text. In *EMNLP*, pages 1266–1276, 2010. 119, 120

D. Lowd and P. Domingos. Efficient Weight Learning for Markov Logic Networks. In *PKDD*, pages 200–211, 2007. 6, 47

X. Luo. On coreference resolution performance metrics. In *HLT/EMNLP*, pages 25–32, 2005. 116

A. McCallum and B. Wellner. Conditional models of identity uncertainty with application to noun coreference. In *NIPS*, pages 905–912, 2004. 113

J. F. Mccarthy and W. Lehnert. Using decision trees for coreference resolution. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 1050–1555, 1995. 68

R. McDonald, K. Hall, and G. Mann. Distributed training strategies for the structured perceptron. In *NAACL2010*, pages 456–464, 2010. 55

R. Mitkov. Factors in anaphora resolution: they are not the only things that matter. a case study based on two different approaches. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 8th Conference of the European Chapter of the Association for Computational Linguistics Workshop on Operational Factors in Practical*, 1997. 68

R. Mitkov, R. Evans, C. Orasan, C. Barbu, L. Jones, and Violeta Sotirova. Coreference and anaphora: developing annotating tools, annotated resources and annotation strategies. In *Proceedings of the Discourse Anaphora and Anaphora Resolution Colloquium*, 2000. 65

R. Mulkar, J. Hobbs, and E. Hovy. Learning from Reading Syntactically Complex Biology Texts. In *Proceedings of the 8th International Symposium on Logical Formalizations of Commonsense Reasoning*, Palo Alto, 2007. 5, 21

M. Murata, H. Isahara, and M. Nagao. Resolution of indirect anaphora in japanese sentences using examples "xnoy(yof x)". In *Proceedings of the ACL 1999 Workshop on Coreference and Its Applications*, 1999. 69, 80

H. Nakaiwa, S. Shirai, S. Ikehara, and T. Kawaoka. Extrasentential resolution of Japanese zero pronouns using semantic and pragmatic constraints. In *Proc. of AAAI 1995 Spring Symposium Series, Empirical Methods in Discourse Interpretation and Generation*. AAAI, 1995. 70

H. T. Ng and R. J. Mooney. Abductive plan recognition and diagnosis: A comprehensive empirical evaluation. In *KR*, pages 499–508, 1992. 56

V. Ng. Shallow semantics for coreference resolution. In *IJCAI*, pages 1689–1694, 2007. 115

V. Ng. Supervised noun phrase coreference research: The first fifteen years. In *Proceedings of the ACL*, pages 1396–1411, 2010. 114

V. Ng and C. Cardie. Identifying anaphoric and non-anaphoric noun phrases to improve coreference resolution. In *Proceedings of 19th COLING, 2002*, pages 1–7, 2002. 68, 69

V. Ng and Claire Cardie. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 104–111, 2001. 68, 75, 77

NLRI, editor. *Bunrui Goi Hyo (in Japanese)*. Shuei Shuppan, 1964. 77

M. Okumura and K. Tamura. Zero pronoun resolution in japanese discourse based on centering theory. In *Proceedings of the 16th conference on Computational linguistics*, pages 871–876, 1996. 68

E. Ovchinnikova. *Integration of World Knowledge for Natural Language Understanding*. Atlantis Press, Springer, 2012. 59, 109, 116, 121

E. Ovchinnikova, N. Montazeri, T. Alexandrov, J. Hobbs, M. McCord, and R. Mulkar-Mehta. Abductive Reasoning with a Large Knowledge Base for Discourse Processing. In *Proceedings of the International Conference on Computational Semantics*, pages 225–234, Oxford, UK, 2011. 5, 21, 39, 109, 116

A. Penas and E. Hovy. Filling knowledge gaps in text for machine reading. In *Proceedings of COLING: Posters*, pages 979–987, 2010. 2, 133

M. Poesio, R. Vieira, and S. Teufel. Resolving bridging references in unresricted text. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pages 1–6, 1997. 69

M. Poesio, T. Ishikawa, S. Schulte Im Walde, R. Vieira, and R. Sinos. Acquiring lexical knowledge for anaphora resolution. In *Proceedings of the 3rd Conference on Language Resources and Evaluation*, pages 1220–1224, 2002. 69

M. Poesio, R. Mehta, A. Maroudas, and J. Hitzeman. Learning to resolve bridging references. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pages 144–151, 2004. 64, 69, 80

S. P. Ponzetto and M. Strube. Exploiting semantic role labeling, wordnet and wikipedia for coreference resolution. In *the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL '06, pages 192–199, 2006. 115

D. Poole. Logic programming, abduction and probability: a top-down anytime algorithm for estimating prior and posterior probabilities. *New Gen. Comput.*, 11(3-4):377–400, July 1993a. 14

D. Poole. Logic Programming, Abduction and Probability: a top-down anytime algorithm for estimating prior and posterior probabilities. *New Generation Computing*, 11(3-4):377–400, 1993b. 5, 21

S. Pradhan, L. Ramshaw, M. Marcus, M. Palmer, R. Weischedel, and N. Xue. CoNLL-2011 shared task. In *CoNLL: shared task*, pages 1–27, 2011. 59, 115, 116

H. Prendinger and M. Ishizuka. First-Order Diagnosis by Propositional Reasoning: A Representation-Based Approach. In *DX*, pages 220–225, 1999. 43

S. Raghavan and R. J. Mooney. Bayesian Abductive Logic Programs. In *Star-AI 10*, pages 82–87, 2010. 6, 14, 16, 38, 47, 56, 60, 61

S. Raghavan, R. J. Mooney, and H. Ku. Learning to "Read Between the Lines" using Bayesian Logic Programs. In *ACL*, pages 349–358, 2012. 131

A. Rahman and V. Ng. Resolving Complex Cases of Definite Pronouns: The Winograd Schema Challenge. In *Proceedings of EMNLP-CoNLL*, pages 777–789, 2012. 108, 115, 122, 133

M. Recasens and E.H. Hovy. BLANC: Implementing the Rand Index for Coreference Evaluation. *Journal of Natural Language Engineering*, 2010. 116, 117

R. Reiter. On closed world data bases. In H. Gallaire and J. Minker, editors, *Logic and Data Bases*, pages 119–40. Plenum Publ. Co., 1978. 32

M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, pages 107–136, 2006. 6, 38, 45, 47, 61

S. Riedel. Improving the Accuracy and Efficiency of MAP Inference for Markov Logic. In *UAI*, pages 468–475, 2008. 36, 42, 46, 132

S. Riedel and J. Clarke. Incremental Integer Linear Programming for Non-projective Dependency Parsing. In *EMNLP*, pages 129–137, 2006. 36

J. A. Robinson. A Machine-Oriented Logic Based on the Resolution Principle. *Journal of the ACM*, 12:23–41, 1965. 9, 22, 23, 32

J. Ruppenhofer, M. Ellsworth, M.R. Petruck, C.R. Johnson, and J. Scheffczyk. FrameNet II: Extended Theory and Practice. Technical report, 2010. 2, 39, 59, 121

E. Santos. A linear constraint satisfaction approach to cost-based abduction. *Artificial Intelligence*, 65 (1):1–27, 1994. 21, 43, 134

E. Santos. Polynomial solvability of cost-based abduction. *Artificial Intelligence*, 86:157–170, 1996. 5

T. Sato and Y. Kameya. New advances in logic-based probabilistic modeling by prism. In Luc Raedt, Paolo Frasconi, Kristian Kersting, and Stephen Muggleton, editors, *Probabilistic Inductive Logic Programming*, volume 4911 of *Lecture Notes in Computer Science*, pages 118–155. Springer Berlin Heidelberg, 2008. 45

S. Schoenmackers, J. Davis, O. Etzioni, and D. Weld. Learning First-order Horn Clauses from Web Text. In *EMNLP*, pages 1088–1098, 2010. 2

L. Shalom and L. H. J. An algorithm for pronominal anaphora resolution. *Computational Linguistics.*, 20(4):535–561, 1994. 68

D. Shen and M. Lapata. Using Semantic Roles to Improve Question Answering. In *Proceeding of EMNLP-CoNLL*, pages 12–21, 2007. 125, 132

S. Sidney and N. J. Castellan. *Nonparametric statistics for the Behavioral Sciences*. McGraw Hill, 1988. 99

P. Singla and P. Domingos. Memory-Efficient Inference for Relational Domains. In *AAAI*, pages 488–493, 2006. 46

P. Singla and P. Domingos. Abductive Markov Logic for Plan Recognition. In *AAAI-11*, pages 1069–1075, 2011. 6, 14, 38, 41, 42, 45, 47, 48, 56, 57, 60, 61

W. M. Soon, H. T. Ng, and C. Y. Lim. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544, 2001a. 64, 68, 69, 75, 77

W. M. Soon, H. T. Ng, and D. C. Y. Lim. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics.*, 27(4):521–544, 2001b. ISSN 0891-2017. 59

M. E. Stickel. A prolog-like inference system for computing minimum-cost abductive explanations in natural-language interpretation. *Annals of Mathematics and Artificial Intelligence*, 4(1):89–105, 1991. 43

M. Strube and C. Muller. A machine learning approach to pronoun resolution in spoken dialogue. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 168–175, 2003. 68

F. M. Suchanek, G. Kasneci, and G. Weikum. YAGO: A Large Ontology from Wikipedia and WordNet. *J. Web Sem.*, 6(3):203–217, 2008. 122

A. Sumida, N. Yoshinaga, and K. Torisawa. Boosting precision and recall of hyponymy relation acquisition from hierarchical layouts in wikipedia. In *Proceedings of the 6h Language Resources and Evaluation Conference*, 2008. 77

C. J. Yu T. Joachims, T. Finley. Cutting-plane training of structural svms. In *Machine Learning*, pages 27–59, 2009. 36

A. Tamaddoni-Nezhad, R. Chaleil, A. Kakas, and S. Muggleton. Application of abductive ilp to learning metabolic network inhibition from temporal data. *Machine Learning*, 64(1-3):209–230, 2006. 43, 46

H. Tamaki and T. Sato. Old resolution with tabulation. In Ehud Shapiro, editor, *Third International Conference on Logic Programming*, volume 225 of *Lecture Notes in Computer Science*, pages 84–98. Springer Berlin Heidelberg, 1986. ISBN 978-3-540-16492-0. 45

N. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995a. 55, 59

V. N. Vapnik. *The Nature of Statistical Learning Theory*. Wiley, 1995b. 101

R. Vieira and M. Poesio. An empirically based system for processing definite descriptions. *Computational Linguistics*, 26(4):539–593, 2000. 65, 70, 81, 102

K. Yamamoto, N. Inoue, Y. Watanabe, N. Okazaki, and K. Inui. Backpropagation Learning for Weighted Abduction (in Japanese). In *IPSJ SIG Technical Reports*, volume 2012-NL-206, 2012. 54

X. Yang, G. Zhou, J. Su, and C. L. Tan. Coreference resolution using competition learning approach. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 176–183, 2003. 68, 75, 77

M. A. Yosef, J. Hoffart, I. Bordino, M. Spaniol, and G. Weikum. Aida: An online tool for accurate disambiguation of named entities in text and tables. In *Proc. of the 37th Intl. Conference on Very Large Databases (VLDB 2011)*, pages 1450–1453, 2011. 122

C. J. Yu and T. Joachims. Learning structural SVMs with latent variables. In *ICML*, 2009. 53, 62

List of Publications

Journal Papers (Refereed)

1. Naoya Inoue, Ryu Iida, Kentaro Inui, and Yuji Matsumoto. Resolving Direct and Indirect Anaphora for Japanese Definite Noun Phrases. Journal of Natural Language Processing, Vol.17, No.1, pp.221-246, January 2010.

2. Ryu Iida, Mamoru Komachi, Naoya Inoue, Kentaro Inui, and Yuji Matsumoto. Annotating Predicate-Argument Relations and Anaphoric Relations: Findings from the Building of the NAIST Text Corpus (in Japanese). Journal of Natural Language Processing, Vol.17, No.2, pp.25-50, April 2010.

International Conference/Workshop Papers (Refereed)

1. Naoya Inoue, Ekaterina Ovchinnikova, Kentaro Inui, and Jerry R. Hobbs. Coreference Resolution with ILP-based Weighted Abduction. In Proceedings of the 24th International Conference on Computational Linguistics, pp. 1291–1308, December 2012.

2. Naoya Inoue and Kentaro Inui. Large-scale Cost-based Abduction in Full-fledged First-order Predicate Logic with Cutting Plane Inference. In Proceedings of the 13th European Conference on Logics in Artificial Intelligence, pp.281-293, September 2012.

3. Naoya Inoue and Kentaro Inui. ILP-based Reasoning for Weighted Abduction. In Proceedings of AAAI Workshop on Plan, Activity and Intent Recognition, pp. 25-32, August 2011.

4. Naoya Inoue, Ryu Iida, Kentaro Inui, and Yuji Matsumoto. Resolving Direct and Indirect Anaphora for Japanese Definite Noun Phrases. In Proceedings of the Conference of the Pacic Association for Computational Linguistics, pp.268-273, September 2009.

5. Kazeto Yamamoto, Naoya Inoue, Yotaro Watanabe, Naoaki Okazaki and Kentaro Inui. Discriminative Learning of First-order Weighted Abduction from Partial Discource Explanations. In Proceedings of the 14th International Conference on Intelligent Text Processing and Computational Linguistics, pp.1291–1380, March 2013.

Awards

1. Annual Meeting Excellent Paper, Association for Natural Language Processing (2012)

2. IPSJ Yamashita SIG Research Award (2012)

3. The 15th Information-Based Induction Sciences Workshops Honorable Mention (2012)

4. IPSJ Certificate of Excellent Master's Thesis (2010)

Other Publications (Not refereed)

1. Naoya Inoue, Kazeto Yamamoto, Yotaro Watanabe, Naoaki Okazaki and Kentaro Inui. Online Large-margin Weight Learning for First-order Logic-based Abduction. In Proceedings of the 15th Information-Based Induction Sciences Workshop, pp.143-150, November 2012. **(Honorable Mention)**

2. Naoya Inoue and Kentaro Inui. Extending ILP-based Abductive Inference with Cutting Plane Inference. In IPSJ SIG Technical Reports, Vol. 2012-NL-208 (5), pp.1-8, September 2012.

3. Naoya Inoue, Kentaro Inui, Ekaterina Ovchinnikova, and Jerry R. Hobbs. Study on Abductive Discourse Processing using Large Knowledge Base (in Japanese). Proceedings of the 18th Annual Meeting of the Association for

Natural Language Processing, pp.119-122, March 2012. **(Annual Meeting Excellent Paper)**

4. Naoya Inoue and Kentaro Inui. An ILP Formulation of Abductive Inference for Discourse Interpretation. In IPSJ SIG Technical Reports, Vol. 2011-NL-203 (3), pp.1-13, September 2011. **(IPSJ Yamashita SIG Research Award)**

5. Naoya Inoue and Kentaro Inui. Toward Plan Recognition in Discourse Using Large-Scale Lexical Resources. Proceedings of the 17th Annual Meeting of the Association for Natural Language Processing, pp. 928-931, March 2011.

6. Naoya Inoue, Ryu Iida, Kentaro Inui, and Yuji Matsumoto. Resolving Direct and Indirect Anaphora in Japanese Texts (in Japanese). Proceedings of Information Processing Society of Japan 50th Anniversary and 72nd National Convention of IPSJ, pp.2-541-542, March 2010. **(IPSJ Certificate of Excellent Master's Thesis)**

7. Naoya Inoue, Ryu Iida, Kentaro Inui, and Yuji Matsumoto. Resolving Direct and Indirect Anaphora for Japanese Definite Noun Phrases (in Japanese). Proceedings of the 15th Annual Meeting of the Association for Natural Language Processing, pp.372-375, March 2009.

8. Kazeto Yamamoto, Naoya Inoue, Yotaro Watanabe, Naoaki Okazaki, and Kentaro Inui. Backpropagation Learning for Weighted Abduction (in Japanese). In IPSJ SIG Technical Reports, Vol.2012-NL-206, May 2012.

9. Jun Sugiura, Naoya Inoue, and Kentaro Inui. Towards Abductive Discourse Relation Recognition (in Japanese). Proceedings of the 18th Annual Meeting of the Association for Natural Language Processing, pp.115-118, March 2012.