Extending ILP-based Abductive Reasoning with Cutting Plane Inference

Naoya Inoue[1,a) Kentaro Inui[1,b)

Abstract: Abduction, inference to the best explanation, is desirable for many natural language processing (NLP) tasks. While recent advances in large-scale world knowledge acquisition warrant applying abduction with large knowledge bases to real-life NLP problems, as of yet no existing approach to abduction has achieved both the efficiency and formal expressiveness necessary to be a practical solution for large-scale reasoning on real-life problems. In this paper, we provide an expressive and efficient solution for large-scale abductive reasoning, extending our prior work on the Integer Linear Programming-based formulation of first-order predicate logic abduction [16], [17]. The key contributions of this paper are the following: (i) we show how Cutting Plane Inference, an iterative optimization strategy developed in Operations Research, can be applied for improving the bottleneck of large-scale first-order logic abduction; (ii) we show the runtime efficiency of our method on a larger and real-life dataset, while existing abductive reasoning systems are evaluated on rather small datasets; (iii) we make the abductive inference engine presented in this paper publicly available.

1. Introduction

Discovering implicit information from natural language discourse is essential to a wide range of NLP tasks, such as Question Answering, Information Extraction and Recognizing Textual Entailment (RTE). A number of NLP components are exploited for processing a variety of discourse phenomena (e.g. anaphora) when inferring implicit information. In the field of computational linguistics, each NLP component has been studied extensively in recent decades; however, less attention has been paid to how to integrate them into a single inference framework.

In this paper, we explore *first-order logic abduction*-based discourse processing as a framework for integrating NLP components. Abduction is inference to the best explanation. Abduction has long been studied in a wide range of contexts from artificial intelligence research to cognitive science. For example, abduction has been viewed as a promising framework for describing the mechanism of human perception [7], [14], [25], [36], etc. The idea is that the declarative nature of abduction enables us to infer the most plausible, implicitly stated information combining several types of inference, and pieces of explicitly observed information.

Abduction-based discourse processing has been studied intensively in the 1980s and 1990s; Hobbs et al. (1993) show that the lowest-cost abductive proof provides the solutions to a broad range of natural language pragmatics problems, such as word sense disambiguation, anaphora, and metonymy resolution. The key advantages of using abduction for discourse processing are twofold:

- abduction-based discourse processing models interdependencies between NLP tasks and identifies the most coherent interpretation to all tasks;
- it discovers plausible and new information by combining heterogeneous inference rules and the pieces of information observed from texts.

While the lack of world knowledge resources hampered applying abduction to real-life problems in the 1980s and 1990s, a number of techniques that acquire world knowledge resources have been developed in the last decade [6], [11], [15], [33], [35], etc. Consequently, several researchers start applying abduction to real-life problems, exploiting large knowledge bases. For instance, inspired by Hobbs et al. (1993), Ovchinnikova et al. (2011) propose an abduction-based natural language processing framework using forty thousands of axioms extracted from the popular ontological resources, WordNet [11] and FrameNet [33]. They evaluate their approach on the real-life natural language processing task of Recognizing Textual Entailment (RTE) [9].

However, in order to apply large-scale abductive inference to real-life problems, we still need to address the following issue: *how to search for the best explanation efficiently*. Abduction is known to be an NP-hard problem in general [5]; this hampers the application of abduction with large world knowledge resources to real-life problems. In fact, Ovchinnikova et al. (1993) report that the Mini-TACITUS abductive reasoning system [22] could not search the entire search space of explanations within 30 minutes in most of the RTE problems in their experiments. Our recent effort, a machine learning-based approach to abductive inference [20], also motivates us to develop more efficient inference engines, because it requires inference as a subroutine. In

[1] Tohoku University,
6-3-09 Aramaki Aza Aoba, Aobaku, Sendai 980-8579
a) naoya-i@ecei.tohoku.ac.jp
b) inui@ecei.tohoku.ac.jp

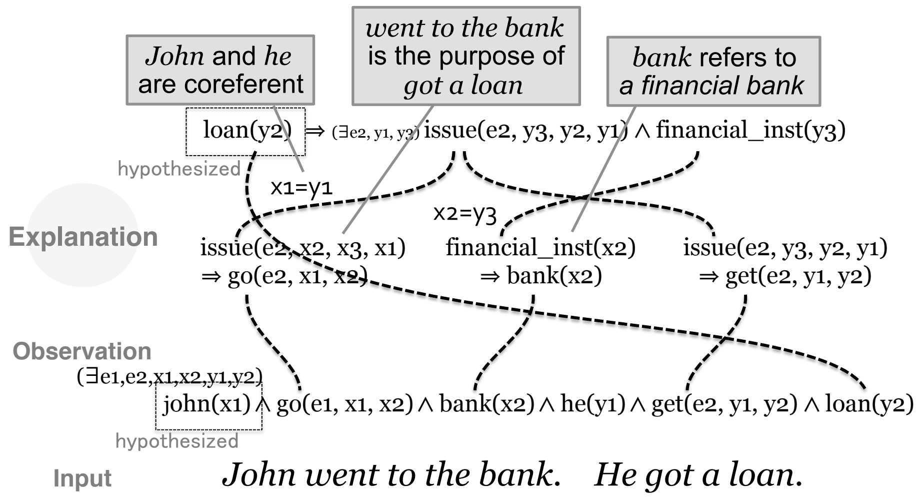


Fig. 1 Example of abductive discourse interpretation.

the literature, many researchers have tried to overcome abduction's inefficiency by a range of methods from approximation to exact inference [8], [18], [26], [34], etc. However, to the best of our knowledge, most of the proposed methods are optimized for propositional logic. It thus requires to transform knowledge bases written in first-order predicate logic (FOPL) to propositional level (i.e. *grounding*) in order to employ these methods. Typically, grounding-based approaches generate a quite huge search space, and does not scale to larger problems.

In this paper, we extend our prior work on the Integer Linear Programming (ILP)-based formulation of first-order logic abduction [16], [17] in order to provide an expressive and scalable solution to abductive inference problems. The proposed method accepts first-order logic as a meaning representation, and is shown to work efficiently on the real-life NLP problem of RTE. The contributions of this paper are the following:

(i) we describe how Cutting Plane Inference (CPI), an iterative optimization strategy developed in Operations Research, can be exploited for making FOPL abductive reasoning tractable;

(ii) we show the runtime efficiency of CPI-based approach by providing evaluation on a large and real-life NLP dataset, while existing abductive reasoning systems [16], [17], [21], [37], etc. are evaluated on rather small datasets;

(iii) the abductive inference engine presented in this paper is made publicly available.

The structure of our paper is as follows. We start with a brief review of abduction and abduction-based discourse processing, taking Hobbs et al. (1993)'s *Interpretation as Abduction* framework as an motivating example (Sec. 2). We give a brief review of our prior work, the ILP-based formulation of abduction (Sec. 3.1), and then show how Cutting Plane Inference makes large-scale FOPL abductive reasoning tractable (Sec. 3.2). We then evaluate the efficiency of our CPI-based framework on a large and real-life problem of natural language processing, RTE (Sec. 4). Finally, we compare our work with existing abductive reasoning approaches (Sec. 5).

2. Background

2.1 Cost-based abduction

Abduction is inference to the best explanation. Formally, logical abduction is defined as follows:

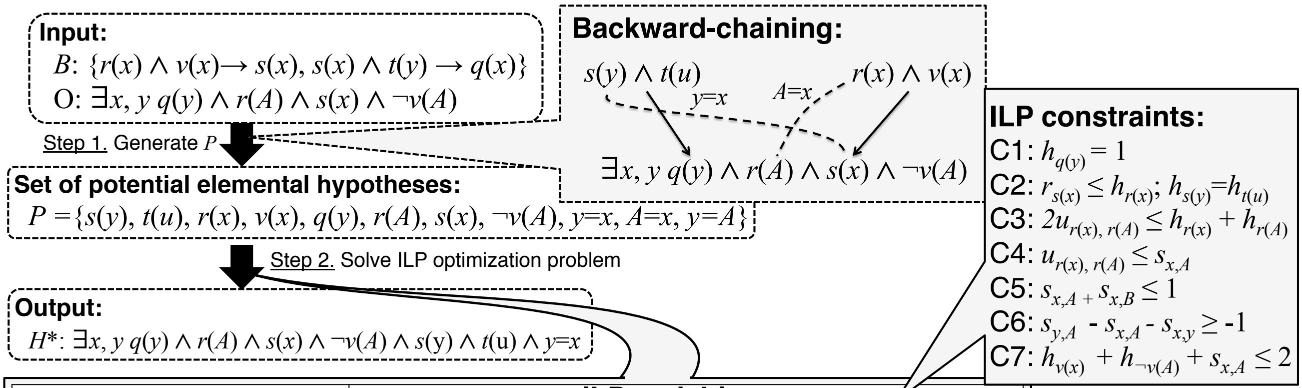
- **Given:** Background knowledge B, and observations O, where both B and O are sets of first-order logical formulas.
- **Find:** A *hypothesis* (or *explanation*) H such that $H \cup B \models O, H \cup B \not\models \perp$, where H is a set of first-order logical formulas. We say that p is *hypothesized* if $H \cup B \models p$, and that p is *explained* if $(\exists q)\ q \rightarrow p \in B$ and $H \cup B \models q$.

Typically, there exist several hypotheses H explaining O. We call each of them a *candidate hypothesis*, and each literal in a hypothesis an *elemental hypothesis*. *Cost-based abduction* (CBA) identifies the minimum-cost explanation H^* among a set $\mathcal{H}$ of candidate explanations. Formally, we find $H^* = \operatorname{argmin}_{H \in \mathcal{H}} cost(H)$, where *cost* is a function $\mathcal{H} \rightarrow \mathbb{R}$, which is called the *cost function*. In the literature, several kinds of cost functions have been proposed, including cost-based and probability-based [7], [14], [27], [29], [37], etc. We elaborate on how abduction can be potentially useful for solving NLP problems in the next section.

2.2 Interpretation as Abduction

Hobbs et al. [14] 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. Figure 1 depicts an example taken from [14], where the coreference relation between *John* and *he*, the intention of *John*, and other implicitly stated information are identified as byproduct of finding an explanation to a given text.

According to [14], one of the important things in abduction-based NLP is that the cost function should be able to evaluate two



Candidate Hypothesis:	ILP variables:												
	$h_{q(y)}$	$h_{r(A)}$	$h_{s(x)}$	$h_{\neg v(x)}$	$h_{s(y)}$	$h_{t(u)}$	$h_{r(x)}$	$h_{v(x)}$	$s_{x,A}$	$s_{x,y}$	$s_{y,A}$	$u_{s(x),s(x)}$	$u_{r(x),r(A)}$
H_1: $q(y) \wedge r(A) \wedge s(x) \wedge \neg v(A)$	1	1	1	1	0	0	0	0	0	0	0	0	0
H_2: $q(y) \wedge r(A) \wedge s(x) \wedge \neg v(A) \wedge s(y) \wedge t(u)$	1	1	1	1	1	1	0	0	0	0	0	0	0
H_3: $q(y) \wedge r(A) \wedge s(x) \wedge \neg v(A) \wedge r(x) \wedge v(x)$	1	1	1	1	0	0	1	1	0	0	0	0	0
H_4: $q(y) \wedge r(A) \wedge s(x) \wedge \neg v(A) \wedge s(y) \wedge t(u) \wedge x=y$	1	1	1	1	1	1	0	0	0	1	0	1	0

Fig. 2 Summary of Inoue and Inui [16], [17]'s ILP-based approach.

types of plausibility of explanations simultaneously: the *correctness* and *informativeness*. The correctness represents how much reliable the contents of information are. For instance, abduction-based NLP systems should be able to judge which inference "*hate*" or "*like*" is better to explain "*holding one's hand*." The informativeness is how specific the information is. In abductive inference, the more backward-chaining is performed, the more specific an explanation becomes. However, we want to stop at some appropriate level because there are often little evidence to support too specific explanations.

To measure the two types of plausibility, Hobbs et al. [14] propose the cost function that gives a penalty for assuming specific and unreliable information but rewards for inferring the same information from different observations. The resulting framework is called *Weighted Abduction*. To the best of our knowledge, Hobbs et al. [14]'s weighted abduction is the only framework that concerns the appropriateness of hypothesis specificity. Because our prior study [16], [17] implements the cost function of Weighted Abduction, the framework proposed in this paper can also accommodate the appropriateness of hypothesis specificity.

3. Extending ILP-based Abductive Reasoning with Cutting Plane Inference

In this section, we extend Inoue and Inui's ILP-based framework of CBA [16], [17] to make the framework more scalable to larger domains. We first give a brief review of Inoue and Inui's ILP formulation, and then propose to apply Cutting Plane Inference to CBA for improving the inefficiency of Inoue and Inui's formulation.

3.1 ILP formulation of CBA

The key idea of Inoue and Inui's formulation [16], [17] is that explanation finding of first-order logic CBA can be regarded as the constrained combinatorial optimization problem of literals and variable substitutions. In principle, this way of problem formulation gives us three benefits. First, we can reduce the search space of candidate hypotheses in comparison to fully grounding approach, because we are able to avoid instantiating FOPL formula with all possible constants. Second, we can exploit the state-of-the-art combinatorial optimization technology developed in Operations Research to find the best explanation. Specifically, our optimization problem can be naturally formulated as the Integer Linear Programming (ILP) problem, which can be efficiently solved by existing ILP solvers. Third, 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.

Let us first give an intuitive description of their approach, using the diagram illustrated in Figure 2. Given an abduction problem (i.e., background knowledge B and observations O), they first create set P of *potential elemental hypotheses*, a set of instantiated literals that are potentially included as constituents of explanations of O (i.e. Step 1 in Figure 2). This procedure is called the *search-space generation*. For enumerating potential elemental hypotheses, they apply *backward-chaining* with axioms in B, and instantiate the body of axioms. For instance, in Figure 2, we add two instantiated literals $s(y)$, $t(u)$ to P, which might be the explanations of $q(y) \in O$, performing backward-chaining on $q(y)$ with axiom $s(x) \wedge t(y) \rightarrow q(x)$. Using the set P, they represent the search space of explanations as an ILP optimization problem as follows.

Hypothesis inclusion: For each $p \in P$, ILP variables $h_p \in \{0, 1\}$ are introduced to represent whether p is hypothesized ($h_p = 1$) or not ($h_p = 0$). For example, H_2 in Figure 2 holds $h_{r(x)} = 1$, where $r(x)$ is included in H_2.

Cost of hypothesis: The cost $cost(H)$ of each candidate hypothesis H is represented by the sum of the costs for $p \in P$ such that p is included in the hypothesis (i.e., $h_p = 1$). Since they follow Hobbs et al. [14]'s weighted abduction, they introduce another ILP variable $r \in \{0, 1\}$ for representing whether p should pay cost ($r_p = 0$) or not ($r_p = 1$). The final objective function of the ILP problem is given by:

$$\textbf{minimize} \quad cost(H) = \sum_{p \in \{p | p \in P, h_p = 1, r_p = 0\}} cost(p), \tag{1}$$

where $cost(p)$ is the cost of a literal p. The cost function can be automatically learned from datasets. For instance, Yamamoto et al. [20] propose a supervised approach to learn the cost function of Weighted Abduction from partially annotated explanations.

The optimization process amounts to Step 2 in Figure 2. Inoue and Inui [16], [17] optimize this objective with the seven types of ILP constraints as shown in Figure 2. In the rest of section, we describe only three of them, which are necessary for the readers to follow the discussion below.

To handle first-order predicate logic, variable substitution must be taken into account to control the unification of elemental hypotheses. For representing the status of unification, they introduce another class of variables $u_{p,q} \in \{0, 1\}$ for each $p, q \in P$, which takes 1 if p is unified with q. Concerning variable substitution, another type of ILP variables s are introduced, where $s_{x,y} = 1$ if x is substituted for y, 0 otherwise. s is symmetric (i.e., $s_{x,y} = s_{y,x}$). In Figure 2, $u_{r(x),r(A)}$ and $s_{x,A}$ are introduced. In H_3, the variables $u_{r(x),r(A)}$, $s_{x,A}$ are set to 1 because $r(x)$ is unified with $r(A)$, and $x = A$ is assumed. Note that unification of $r(x)$ with $r(A)$ is allowed only if x are substituted with A. In addition, the substitution relation must be transitive (e.g. $y = A$ must hold if $x = y$ and $x = A$ hold). For keeping those consistency, they impose two ILP constraints:

Constraint 4[*1]**:** Two literals $q(x_1, x_2, ..., x_n) \equiv q(\mathbf{x})$ and $q(y_1, y_2, ..., y_n) \equiv q(\mathbf{y})$ are allowed to be unified (i.e., $u_{q(\mathbf{x}),q(\mathbf{y})} = 1$) only if all variable substitutions x/y involved in the unification are activated (i.e., $s_{x_i,y_i} = 1$ for all $i \in \{1, 2, ..., n\}$). This can be expressed as:

$$n \cdot u_{q(\mathbf{x}),q(\mathbf{y})} \leq \sum_{i=1}^{n} s_{x_i,y_i} \tag{2}$$

In Figure 2, the constraint $u_{r(x),r(A)} \leq s_{x,A}$ is generated since x needs to be substituted for A when $r(x)$ and $r(A)$ are unified.

Constraint 6: s is transitive; namely $s_{x,z}$ must be 1 if $s_{x,y} = 1$ and $s_{y,z} = 1$. This can be expressed as the following constraints[*2]:

$$s_{x,z} - s_{x,y} - s_{y,z} \geq -1 \tag{3}$$
$$-s_{x,z} + s_{x,y} - s_{y,z} \geq -1 \tag{4}$$
$$-s_{x,z} - s_{x,y} + s_{y,z} \geq -1 \tag{5}$$

They generate $O(n^3)$ transitivity constraints, where n is the number of logical terms. As the reader will see in Sec. 4, this makes inference intractable in large-scale inference. To handle logical negation, [16] introduce the following constraint.

Constraint 7: Two literals $q(x_1, x_2, ..., x_n) \equiv q(\mathbf{x})$ and $\neg q(y_1, y_2, ..., y_n) \equiv \neg q(\mathbf{y})$ cannot be both hypothesized ($h_{q(\mathbf{x})} = 1$ and $h_{\neg q(\mathbf{y})} = 1$) if variable substitutions x_i/y_i are activated ($s_{x_i,y_i} =$ 1) for all $i \in \{1, 2, ..., n\}$. This can be expressed as follows:

$$h_{q(\mathbf{x})} + h_{\neg q(\mathbf{y})} + \sum_{i=1}^{n} s_{x_i,y_i} \leq 1 + n. \tag{6}$$

Note that the case where $\mathbf{x} = \mathbf{y}$ reduces to $h_{q(\mathbf{x})} + h_{\neg q(\mathbf{x})} \leq 1$. 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.

Algorithm 1 CPI4CBA(Background Knowledge **B**, Observation **O**)

$(\Psi, I) \leftarrow$ createBaseILP(B, O)
repeat
  $S \leftarrow$ solveILP(Ψ, I); $V \leftarrow \{\}$
  for $(x, y) \in$ unifiedTerms(S) **do**
    for $z \in$ termsUnifiableWith$(x) \cup$ termsUnifiableWith(y) **do**
      if ($s_{x,z} = 0$ and $s_{y,z} = 1$) or ($s_{x,z} = 1$ or $s_{y,z} = 0$) **then**
        $V \leftarrow V \cup \{-s_{x,y} - s_{x,z} + s_{y,z} \geq -1, -s_{x,y} + s_{x,z} - s_{y,z} \geq -1\}$
      end if
    end for
  end for
  $I \leftarrow I \cup V$
until $V \neq \phi$

3.2 Cutting plane inference for CBA

The major drawback of the presented ILP formulation is that it generates a large number of transitivity constraints and negation-handling constraints. Consequently, it often makes inference intractable (see Sec. 4 for empirical evidence).

Our solution to this problem is that "all the transitivity constraints may not be violated all at once; so we gradually optimize and add transitivity constraints *if violated* in an iterative manner." More formally, we propose to apply *Cutting Plane Inference (CPI)*[*3] to the CBA. CPI is an exact inference optimization technique that is originally developed for solving large linear programming (LP) problems in Operations Research [10]. CPI has been successfully applied to a wide range of constrained optimization problems from probabilistic deductive inference problems [31] to machine learning problems [19], where constraints are very large [2], [19], [31], [32], etc. To the best of our knowledge, however, our work is the first successful work to apply CPI to abductive reasoning 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 transitivity constraints dynamically when transitiveness of unification is violated (e.g. $x = y \wedge y = z \wedge z \neq x$). The iteration terminates if there is no violated unification transitivity. The pseudo-code is given in

[*1] The numbers of constraints correspond to the numbers presented in [17].
[*2] Inoue and Inui [17] introduce the form of inequality $s_{x,y} + s_{y,z} \leq 2 \cdot s_{x,z}$ as transitivity constraints. However, this constraint does not appropriately represent transitivity; thus we replace them with inequalities (3)–(5).
[*3] [16] propose equivalence cluster-based representation of variable unification to improve the inefficiency of transitivity constraints. We will compare cutting plane inference-based approach with the equivalence cluster-based approach in future work.

Algorithm 1. In line 1, we first create an ILP optimization problem without transitivity constraints (i.e. Constraint 6), where Ψ denotes a set of ILP variables, and I denotes a set of ILP constraints. In line 2–12, we repeat: checking consistency of unification transitiveness, adding constraints for violated transitiveness, and re-optimizing. In line 3, we find the solution S for the current ILP optimization problem. Then, for each pair (x, y) of logical atomic terms unified in the solution S (line 4), find the logical term z which is unifiable with x or y (line 5). If the transitive relation x, y with respect to z is violated (i.e. $s_{x,z} = 0 \wedge s_{y,z} = 1$ or $s_{x,z} = 1 \wedge s_{y,z} = 0$), then we generate constraints for preventing this violation, and keep it in set V of constraints (line 6–8). Finally, we again perform an ILP optimization with newly generated constraints (line 11 and 3). The iteration ends when there is no violated transitiveness (line 12).

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. 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 existing abductive reasoning systems, including Inoue and Inui [16], [17]'s system, are evaluated on rather small, and/or artificial datasets [21], [29], [37], etc, our evaluation takes a real-life, much larger datasets. In our experiments, we compare our system with the systems [3], [21], [37] based on Markov Logic Networks (MLNs) [30]. For our experiments, we have used a 12-Core Opteron 6174 (2.2GHz) 128 GB RAM machine. We used Gurobi Optimizer[*4], which is an efficient ILP solver. It is commercial but an academic license is freely available.

4.1 Settings

STORY: For this setting, we have used Ng and Mooney (92)'s story understanding dataset, which is widely used for evaluation of abductive plan recognition systems [16], [21], [29], [37]. In this task, we need to abductively infer the top-level plans of characters from actions which are represented by the logical forms (e.g. $\mathit{getting\_off}(\mathit{Getoff16}) \wedge \mathit{agent\_get\_off}(\mathit{Getoff16}, \mathit{Fred16}) \wedge \mathit{name}(\mathit{Fred16}, \mathit{Fred})$). The dataset consists of 50 plan recognition problems and 107 background Horn clauses (e.g. $\mathit{go\_step}(r, g) \wedge \mathit{going}(g) \rightarrow \mathit{robbing}(r)$). The dataset contains on average 12.6 literals in observed logical forms. To make the predicates representing top-level plans (e.g. shopping, robbing) disjoint, we generated 73 disjointness axiom by using the formulation[*5] described in Sec. 3.1. Regarding a cost function, we followed Hobbs et al. [14]'s weighted abduction theory. For each axiom, we have set each weight so that the sum of the axiom weights equal to 1.2 (e.g. $\mathit{inst\_shopping}(s)^{0.6} \wedge \mathit{store}(t, s)^{0.6} \rightarrow \mathit{shopping\_place}(t)$).

RTE: For observations (input), we employed the second challenge of RTE dataset[*6]. 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 [13] using the Boxer semantic parser [4]. The number of literals in observations is 29.6 literals on average. For background knowledge, we have extracted 289,655 axioms[*7] from WordNet 3.0 [11], and 7,558 axioms from FrameNet 1.5 [33] following [24]. In principle, the WordNet knowledge base contains several kinds of lexical relations between words, such as IS-A, ontological relations (e.g. $\mathit{dog}(x) \rightarrow \mathit{animal}(x)$). 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: $\mathit{Giving}(e_1, x_1, x_2, x_3)^{1.3} \wedge \mathit{donor}(e_1, x_1)^{0.1} \wedge \mathit{recipient}(e_1, x_2)^{0.2} \wedge \mathit{theme}(e_1, x_3)^{0.1} \rightarrow \mathit{give}(e_1, x_1, x_3) \wedge \mathit{to}(e_2, e_1, x_2)$. We again followed Hobbs et al. (1993)'s weighted abduction theory for calculating the cost of hypothesis. We calculated the costs by following Ovchinnikova et al. (2011) in this setting.

4.2 Results and discussion

The reasoner was given a 2-minute time limit for each inference step (i.e. search-space generation and ILP optimization). In Table 1, we show the results of each setting for two inference method in Table 1: (i) *IAICBA*: the inference method without CPI (i.e. Inoue and Inui's system [16]), 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. 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 total number of constraints considered in the end, including the constraints added by CPI. The number marked by Δ indicates the

*4 http://www.gurobi.com/

*5 For example, we generate $h_{robbing(x)} + h_{shopping(y)} + s_{x,y} \leq 2$.
*6 http://pascallin.ecs.soton.ac.uk/Challenges/RTE2/
*7 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 1 The results of averaged inference time in **STORY** and **RTE**.

number of constraints that are added during CPI (i.e. how many times line 7 in Algorithm 1 executed).

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 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 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 76.9% in "ILP inf" for Depth = ∞ 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. 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 = ∞. 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. First, we immediately see that the proposed method is more efficient than Inoue and Inui [17]'s formulation (i.e. IAICBA). Regarding the MLN-based systems [3], [21], [37], our results are comparable, or more efficient than the existing systems. For the **STORY** setting, Singla and Mooney (2011) report the results of two systems with an exact inference technique using CPI for MLNs [31]: (i) Kate and Mooney (2009)'s approach: 2.93 seconds, and (ii) Singla and Mooney (2011)'s approach: 0.93 seconds[*8]. 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)[*9]. On the other hand, our method solves 76.9% of all the problems, where sub-optimal solutions are still available for the rest of 21.5%, and it takes only 0.84 seconds for search-space generation, and 11.73 seconds for ILP inference.

5. Related work

A number of efficient methods for solving cost-based abduction have been proposed [1], [8], [12], [18], [28], [34], etc; however, most of them focus on improving the inefficiency of propositional logic-based abduction. Although propositionalization techniques are available for applying these methods to FOPL abduction, it will lead to an exponential growth of ground instances. Hence they would not scale to larger problems for performing FOPL abduction with large knowledge bases, as discussed in Sec. 3.1.

One important work for FOPL abduction is Inoue and Inui [16], [17]'s approach formulating first-order predicate logic abduction as an ILP optimization problem. It supports FOPL as a meaning representation, and provides a scalable solution. The key idea to make FOPL inference tractable is that they avoid expanding first-order logic formulas with all possible bindings, and formulate the task of FOPL inference as the constrained combinatorial optimization problem of literals and variable substitutions. However, as mentioned in Sec. 3.1, this formulation still has a

*8 This is the result of MLN-HC in [37]. 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.

*9 They used 56,000 FrameNet axioms in the experiments, while we used 289,655 WordNet axioms and 7,558 FrameNet axioms.

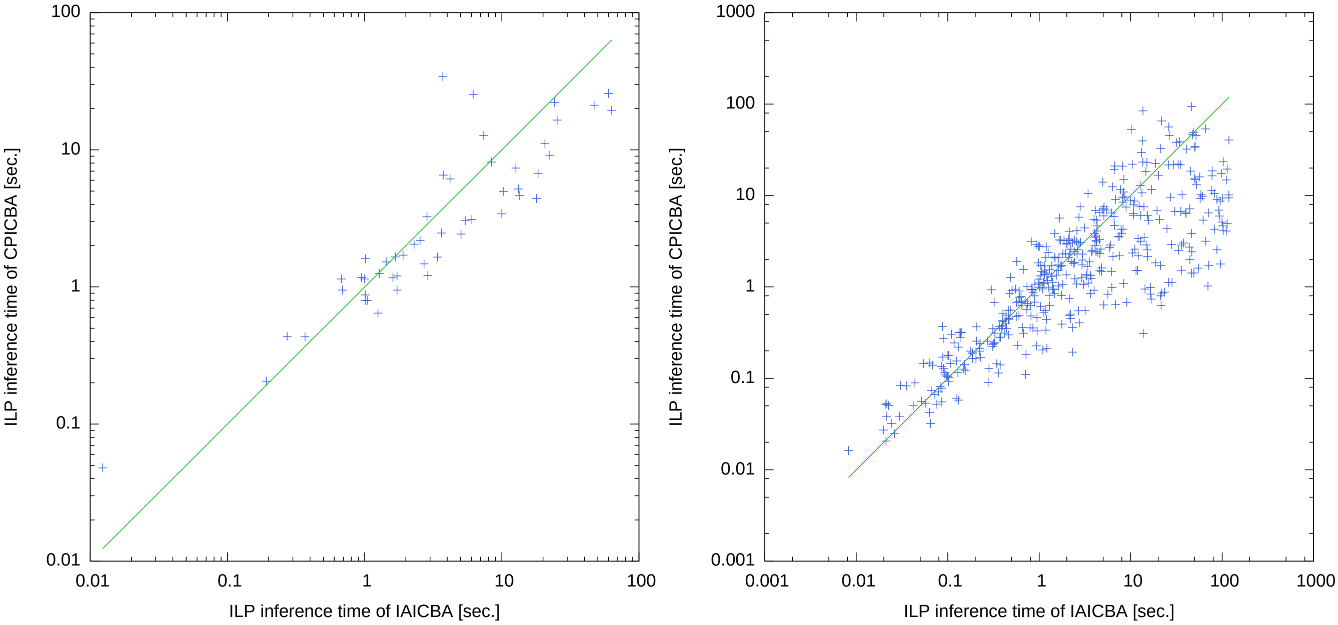


Fig. 3 Runtime comparison between IAICBA and CPI4CBA (logarithmic scale). The left figure shows the results of STORY dataset, and the right figure shows the results of RTE datasets.

significant drawback for large-scale reasoning on real-life problems: the combinatorial growth of transitivity constraints which arises from support for FOPL (see Sec. 3.2). Our work successfully overcomes the significant drawback by employing Cutting Plane Inference-based approach.

Recently, researchers in the field of Statistical Relational Learning have been emulated abductive inference [3], [21], [37], etc. on Markov Logic Networks (MLNs) [30]. MLNs provide full support of first-order predicate logic and the 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 [21] 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. As for learning, we are also working on developing a machine learning-based approach for cost assignments (see [20] for further detail).

6. Conclusion

We have proposed an efficient ILP-based formulation for cost-based abduction in first-order predicate logic, extending Inoue and Inui [16], [17]'s approach. Compared to prior work, our method is more expressive and efficient. Although FOPL 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 a real-life problem (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 paper is made publicly available.

In future work, we plan to apply Cutting Plane Inference to both the search-space generation and ILP inference, repeating the generation of potential elemental hypotheses and ILP optimization interactively, as in cutting plane MAP inference in MLNs [31]. We are also planning to develop a machine learning-based approach for automatic tuning of cost functions. Our group has already achieved neural networks-based approach to learn the cost function of Weighted Abduction [20], but we are also exploring another direction: to generalize the cost function as a weighted linear feature function, and then apply a standard linear training algorithm such as perceptrons. We will then evaluate the abduction-based framework in terms of the prediction accuracy on real-life tasks. We intend to apply abduction to co-reference resolution as a first step.

Acknowledgments

This work was partially supported by Grant-in-Aid for JSPS Fellows (22-9719) and Grant-in-Aid for Scientific Research (23240018).

References

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

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

[3] 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, Oxford, UK, 2011.

[4] J. Bos. Wide-Coverage Semantic Analysis with Boxer. In *STEP*, pages 277–286, 2008.

[5] T. Bylander, D. Allemang, M. C. Tanner, and J. R. Josephson. The computational complexity of abduction. *Artificial Intelligence*, 49(1-3):25–60, may 1991.

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

[7] E. Charniak and R. P. Goldman. A Probabilistic Model of Plan Recognition. In *AAAI*, pages 160–165, 1991.

[8] 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.

[9] I. Dagan, B. Dolan, B. Magnini, and D. Roth. Recognizing textual entailment: Rational, evaluation and approaches - Erratum. *NLE*, 16(1):105, 2010.
[10] G. B. Dantzig, R. Fulkerson, and S. M. Johnson. Solution of a large-scale traveling salesman problem. *Operations Research*, 2(4):393–410, 1954.
[11] C. Fellbaum, editor. *WordNet: an electronic lexical database*. MIT Press, 1998.
[12] 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.
[13] J. R. Hobbs. Ontological promiscuity. In *ACL*, pages 61–69, Chicago, Illinois, 1985.
[14] J. R. Hobbs, M. Stickel, P. Martin, and D. Edwards. Interpretation as abduction. *Artificial Intelligence*, 63:69–142, 1993.
[15] D. Hovy, C. Zhang, E. Hovy, and A. Penas. Unsupervised discovery of domain-specific knowledge from text. In *ACL*, pages 1466–1475, 2011.
[16] N. Inoue and K. Inui. An ILP Formulation of Abductive Inference for Discourse Interpretation. In *IPSJ SIG Technical Reports*, volume 2011-NL-203.
[17] N. Inoue and K. Inui. ILP-Based Reasoning for Weighted Abduction. In *AAAI Workshop on Plan, Activity and Intent Recognition*, 2011.
[18] 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.
[19] T. Joachims, T. Finley, and C. J. Yu. Cutting-plane training of structural svms. In *Machine Learning*, pages 27–59, 2009.
[20] K. Yamamoto and N. Inoue and Y. Watanabe and N. Okazaki and K. Inui. Backpropagation Learning for Weighted Abduction (in Japanese). volume 2012-NL-206, 2012.
[21] R. J. Kate and R. J. Mooney. Probabilistic Abduction using Markov Logic Networks. In *PAIRS*, 2009.
[22] R. Mulkar, J. Hobbs, and E. Hovy. Learning from Reading Syntactically Complex Biology Texts. In *Commonsense*, Palo Alto, 2007.
[23] H. T. Ng and R. J. Mooney. Abductive Plan Recognition and Diagnosis: A Comprehensive Empirical Evaluation. In *KR*, pages 499–508, 1992.
[24] E. Ovchinnikova, N. Montazeri, T. Alexandrov, J. R Hobbs, M. McCord, and R. Mulkar-Mehta. Abductive Reasoning with a Large Knowledge Base for Discourse Processing. In *IWCS*, pages 225–234, Oxford, UK, 2011.
[25] S. Espinosa Peraldi, A. Kaya, S. Melzer, R. Möller, and M. Wessel. Multimedia Interpretation as Abduction. In *Int'l Workshop on Description Logics*, 2007.
[26] 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, 1993.
[27] D. Poole. Probabilistic Horn abduction and Bayesian networks. *Artificial Intelligence*, 64 (1):81–129, 1993.
[28] H. Prendinger and M. Ishizuka. First-Order Diagnosis by Propositional Reasoning: A Representation-Based Approach. In *DX*, pages 220–225, 1999.
[29] S. Raghavan and R. J. Mooney. Bayesian Abductive Logic Programs. In *STARAI*, pages 82–87, 2010.
[30] M. Richardson and P. Domingos. Markov logic networks. *ML*, pages 107–136, 2006.
[31] S. Riedel. Improving the Accuracy and Efficiency of MAP Inference for Markov Logic. In *UAI*, pages 468–475, 2008.
[32] S. Riedel and J. Clarke. Incremental Integer Linear Programming for Non-projective Dependency Parsing. In *EMNLP*, pages 129–137, 2006.
[33] J. Ruppenhofer, M. Ellsworth, M.R. Petruck, C.R. Johnson, and J. Scheffczyk. FrameNet II: Extended Theory and Practice. Technical report, Berkeley, USA, 2010.
[34] E. Santos. A linear constraint satisfaction approach to cost-based abduction. *Artificial Intelligence*, 65 (1):1–27, 1994.
[35] S. Schoenmackers, J. Davis, O. Etzioni, and D. Weld. Learning First-order Horn Clauses from Web Text. In *EMNLP*, pages 1088–1098, 2010.
[36] Murray Shanahan. Perception as Abduction: Turning Sensor Data Into Meaningful Representation. *Cognitive Science*, 29(1):103–134, 2005.
[37] P. Singla and P. Domingos. Abductive Markov Logic for Plan Recognition. In *AAAI*, pages 1069–1075, 2011.