Boosting the Efficiency of First-Order Abductive Reasoning Using Pre-estimated Relatedness between Predicates

Kazeto Yamamoto, Naoya Inoue, Kentaro Inui, Yuki Arase and Jun'ichi Tsujii

***Abstract*—Abduction is inference to the best explanation. While abduction has long been considered a promising framework for natural language processing (NLP), its computational complexity hinders its application to practical NLP problems. In this paper, we propose a method to predetermine the semantic relatedness between predicates and to use that information to boost the efficiency of first-order abductive reasoning. The proposed method uses the estimated semantic relatedness as follows: (i) to block inferences leading to explanations that are semantically irrelevant to the observations, and (ii) to cluster semantically relevant observations in order to split the task of abduction into a set of non-interdependent subproblems that can be solved in parallel. Our experiment with a large-scale knowledge base for a real-life NLP task reveals that the proposed method drastically reduces the size of the search space and significantly improves the computational efficiency of first-order abductive reasoning compared with the state-of-the-art system.**

***Index Terms*—Natural language processing, logical inference, abduction.**

I. Introduction

Abduction is inference from a given set of observations to the best explanation about why those observed events happened. This mode of inference has long been applied to a range of AI tasks including text/story understanding and plan/intention recognition [1]-[6].

An epoch-making study in this line of research can be seen in a paper in *Artificial Intelligence* by Hobbs *et al.* [1]; they demonstrate that a wide range of subtasks in the understanding of natural language can be uniformly formulated as abductive reasoning. Let us take an example from their paper: *John went to the bank. He got a loan.* Given this text as input, it is assumed that the model of Hobbs et al. semantically parses it to obtain a logical form, which consists of a flat conjunctive set of observed literals, as shown at the bottom of Fig. 1. The model then uses a knowledge base (i.e., a collection of axioms representing linguistic and common-sense knowledge) to search for the best explanation for the given observations. As a by-product of this abductive reasoning, the model obtains an interpretation of the given text, which includes the coreference relation between *John* and *he*, and the purpose-means relation between *get a loan* and *went into the bank.*

This way of formulating intelligent inference has several distinct advantages. First, it provides a uniform framework for integrating subtasks of multiple levels of abstraction; in the above example, finding the best explanation jointly resolves the coreference relation, the discourse relation, and the word-sense ambiguity. Second, the declarative nature of abduction allows us to abstract away from the procedural process of inferences. When multiple levels of interdependent subtasks are involved, it is often crucially difficult to predetermine the optimal order in which to solve the problems. This difficulty can be avoided by using joint inference. In spite of these promising properties, however, the abduction-based approaches to text/story understanding and plan/intention recognition have never produced significant positive evidence that supports their effectiveness in real-life problems.

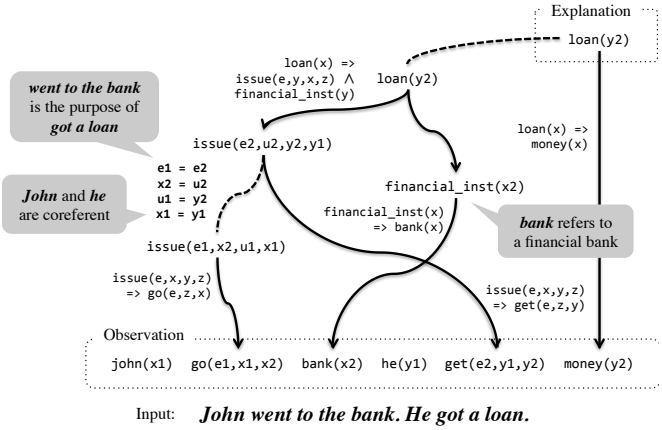


Fig. 1. An example of discourse understanding with abductive reasoning.

One strong reason for this failure has been lack of knowledge. As in other approaches, the bottleneck for applying abduction to practical problem settings in the 1980s and 1990s was the acquisition of knowledge. However, this problem has now been at least partly resolved by recent remarkable advances in the automatic acquisition of linguistic and common-sense knowledge from large-scale text data and the Internet [7], [8]. As a result of these efforts, a number of Web-scale structured and formalized knowledge bases are publicly available [9].

Another big issue is the computational cost of abductive reasoning. Abduction on first-order logic (FOL) or similarly expressive languages is computationally expensive, and thus substantial improvements are necessary in order to make it practical for real-life problems. For this purpose, Inoue and Inui [10], [11] have recently proposed encoding abductive

Manuscript received October 10, 2014; revised December 11, 2014. This work was supported in part by the Grant-in-Aid for JSPS Fellows (22-9719) and Grant-in-Aid for Scientific Research (23240018).

Kazeto Yamamoto, Naoya Inoue, and Kentaro Inui are with Tohoku University, Japan (e-mail: {kazeto,naoya-i,inui}@cl.ecei.tohoku.ac.jp).

Yuki Arase is with Osaka University, Japan (e-mail: arase@ist.osaka-u.ac.jp).

Jun'ichi Tsujii is with Microsoft Research Asia, China (e-mail: jtsujii@microsoft.com).

inference on FOL into a problem of integer linear programming (ILP) and showed that their method significantly improves computational efficiency for a knowledge base containing hundreds of thousands of axioms representing both linguistic and common-sense knowledge. However, the problem of computational cost has not yet been fully solved. The search space for the method of Inoue and Inui still grows exponentially with the size of the knowledge base.

Given this background, in this paper, we explore two methods for improving the computational efficiency of first-order abductive inference. We first explore a method that uses an A* search to reduce the size of the search space. We then explore a method for dividing a given problem into independent subproblems that can be solved in parallel. In our experiments, we show that our system is several tens times as efficient as the state-of-the-art abductive reasoner.

This paper is organized as follows. We first give a brief review of related work. We then show our approach. Afterwards, we demonstrate the efficiency of our methods and compare them with the state-of-the-art system. Finally, we discuss areas of potential future work.

II. Background

A. Abduction

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

Given: Background knowledge B and observations O, where B is a set of Horn clauses on FOL and O is a conjunction of FOL literals.

Find: A *hypothesis* (*explanation*) H such that $H \cup B \models O$, $H \cup B \not\models \bot$, where H is a conjunction of first-order literals.

Typically, there are several hypotheses H that explain O. We call these the *candidate hypotheses*, each literal in a candidate hypothesis is an *elemental hypothesis*, and each literal in possible candidate hypotheses is called a *potential elemental hypothesis*. A candidate hypothesis is a subset of the potential elemental hypotheses, and we can regard the potential elemental hypotheses as defining the search space of the solution.

The goal of abduction is to find the best hypothesis $\hat{H}$ among the candidate hypotheses by using a specific evaluation measure. We call $\hat{H}$ the *solution hypothesis*. Formally, the solution hypothesis is defined as follows:

$$\hat{H} = \arg\max_{H \in \mathbb{H}} E(H) \quad (1)$$

where $\mathbb{H}$ is a set of possible candidate hypotheses, and E is a function $H \to \mathbb{R}$ that evaluates the plausibility of each candidate hypothesis. Here, we assume that $E(H)$ returns $-\infty$ if $H \cup B \not\models \bot$, and we call this the *evaluation function*. In the literature, several kinds of evaluation functions have been proposed [1], [6], [11]-[13].

As noted, each candidate hypotheses can be regarded as a subset of the potential elemental hypotheses. Potential elemental hypotheses are generated by applying the following two operations to the observations and the potential elemental hypotheses being generated:

- **Backward chaining:** Assuming an axiom $p_1(x) \wedge p_2(x) \wedge \cdots \wedge p_n(x) \Longrightarrow q(x) \in B$ and a literal $q(a)$, this operation hypothesizes new literals $\{p_i(a)\}_{i=1}^{n}$ and adds them to the potential elemental hypotheses.
- **Unification:** This operation unifies two literals that have the same predicate and makes the assumption that each term of a literal is equal to the corresponding term of the other literal. For example, given $O = p(x) \wedge p(y) \wedge q(y)$, a candidate hypothesis $H = (x = y)$ is created.

For example, given the knowledge base shown in Table I, let us consider creating the potential elemental hypotheses for the observation $O = \{p_2(a) \wedge p_6(b,c) \wedge p_7(d)\}$. Applying backward chaining and unification to the potential elemental hypotheses as shown in Figure 2, we can obtain the potential elemental hypotheses $P = \{p_1(b)^1 \wedge p_1(b)^2 \wedge p_2(b) \wedge p_3(b) \wedge p_4(u_2,u_1) \wedge p_5(u_1) \wedge p_8(c) \wedge p_9(c) \wedge (a = d)\}$. In Fig. 2, a solid arrow indicates backward chaining, a dotted line indicates unification, and the terms in the gray boxes represent the IDs of the axioms used for the corresponding backward chaining.

TABLE I: A Knowledge Base for an Example

ID	Axiom	ID	Axiom
a_1	$p_1(x) \Rightarrow p_2(x)$	a_5	$p_2(x) \Rightarrow p_7(x)$
a_2	$p_1(x) \Rightarrow p_3(x)$	a_6	$p_5(y) \Rightarrow p_7(x)$
a_3	$p_4(x,y) \Rightarrow p_5(y)$	a_7	$p_8(y) \Rightarrow p_6(x,y)$
a_4	$p_3(x) \Rightarrow p_6(x,y)$	a_8	$p_9(x) \Rightarrow p_8(x)$

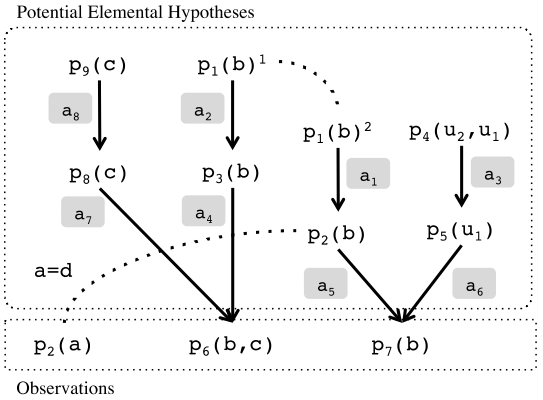


Fig. 2. An example of potential elemental hypotheses.

B. Previous Work for Efficient Abduction

Abductive inference is an NP-hard problem, and so the computational cost increases exponentially with increases in the knowledge base; this is a big problem. The studies that have addressed this issue can be classified roughly into two groups.

The first includes those methods that emulate abduction by using a framework for deduction [6], [13], [14]. For example, Singla and Domingos (2011) proposed a method that emulates abduction on Markov logic networks (MLNs) [15]. However, although these methods can make use of efficient algorithms for the target framework, they are not very efficient [14]. The reason of this is that the grounding, i.e., the process that converts the knowledge base or observations in the first-order logic into propositional logic, causes the knowledge base to increase explosively.

The second includes those methods that formulate abduction as the problem of finding the best subset of the potential elemental hypotheses, and then uses another optimization algorithm to search the subset of potential

elemental hypotheses that corresponds to the solution hypothesis. For example, Inoue and Inui proposed a method to formulate abductive reasoning as a problem of integer linear programming (ILP) without grounding [10], [11]. With this method, a drastic improvement was achieved by the efficiency of the lifted inference and by using an efficient optimization algorithm in an external ILP solver. Inoue and Inui (2012) reported that this approach is much faster than the MLN-based framework discussed above [11], which had been the state of the art before being replaced by this method.

III. Efficient Abduction with Relatedness between Predicates

A. Basic Strategy

We begin by discussing the optimality of the solution obtained by the abduction. In abductive reasoning, because the search space of the solution can increase without limit, obtaining the global optimal solution by abductive reasoning is expensive. Therefore, in practice, it is the local, not the global, optimal solution that is sought; that is, we seek the best hypothesis within some limited search space and regard it as the best explanation. In the work of Inoue and Inui [10], [11], a parameter $depth_{\max}$ was defined to be a natural number, and the potential elemental hypotheses consist of those elemental hypotheses that can be hypothesized through less than $depth_{\max}$ backward chainings. A larger $depth_{\max}$ indicates a higher probability that the solution is a global optimum and a correspondingly higher computational cost. The optimality of the solution and its computational cost both depend on the size of the search space of the solution. In this paper, we aim to reduce the size of the search space (i.e., the number of potential elemental hypotheses) while maintaining the optimality of the solution.

In abduction, the evaluation functions are generally defined so that the better a hypothesis is considered to be, the greater the probability of the assumptions included in the hypothesis and the more observations it explains. For example, given the knowledge base shown in Table I and an observation $O = \{p_6(a,b) \wedge p_7(c)\}$, let us consider the three hypotheses shown in Fig. 3. Here, the hypothesis (b) is less optimal than hypothesis (a), because (b) includes more hypothesized literals than (a) but explains the same number of observations. On the other hand, since hypothesis (c) explains as many observations as (a) with fewer literals, (c) is considered to be better than (a). More formally, the evaluation functions E generally have the following properties:

1) Given a candidate hypothesis H and an operation of backward chaining c, $E(H) \geq E(H \cap c)$ is satisfied.
2) A candidate hypothesis H and an operation of unification u that satisfy $E(H) \leq E(H \cap u)$ can exist.

Supposing that the evaluation function that we employ has these properties, then we can reduce the number of potential elemental hypotheses by canceling the backward chainings that do not result in unification.

B. Heuristic Pre-estimation of the Distance between the Literals

In order to estimate whether the backward chaining will result in unification, it is necessary to know which literals can be hypothesized from each observation and the plausibility of each literal. Here, we define the function *hed*(*p*,*q*), which provides the semantic relatedness between a literal *p* and a literal *q*. We call the return value of *hed*(*p*,*q*) the *heuristically estimated distance* (*H.E.D.*) between *p* and *q*.

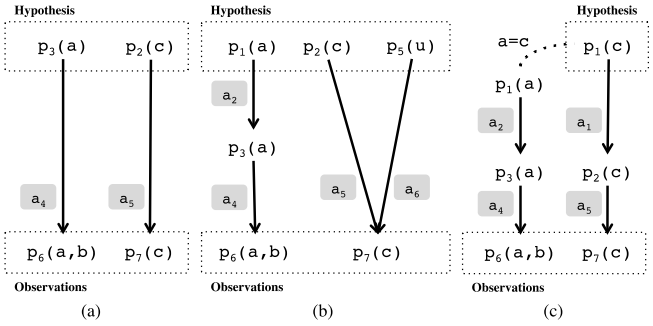


Fig. 3. An example of the basic strategy.

The necessary conditions of *hed*(*p*,*q*) and H.E.D. are as follows. First, they must express the semantic relatedness between *p* and *q*. In other words, the more easily the relevance between two literals can be inferred, the higher the H.E.D. between them. Second, *hed*(*p*,*q*) must be admissible for use in an A* search, so that it can be employed as a heuristic for the cost, as in Section III.C. Third, the computational cost for obtaining a return value from *hed*(*p*,*q*) should be as small as possible. For the third condition, we pre-estimate all of the H.E.D.s and store them in a database. Thus, the function *hed*(*p*,*q*) only has to load values from memory. Since the size of the database of H.E.D.s increases as the definition of *hed*(*p*,*q*) becomes more complex, we have to consider the balance between efficiency and the expressiveness of the H.E.D.s.

Therefore, we define this function as the heuristic distance between the predicates of the literals with the abstraction of the conjunctions of the antecedents of each of the axioms. Formally, *hed*(*p*,*q*) is defined as follows:

$$hed(p,q) = \min_{H \in \{H \mid H \cup B^* \models \{\rho(p) \wedge \rho(q)\}\}} \sum_{a \in A_H} \delta(a) \quad (2)$$

$$B^* = \bigcup_{p_1 \wedge \ldots \wedge p_n \Rightarrow q \in B} \left[\bigcup_{i=1}^{n} \rho(p_i) \Rightarrow \rho(q) \right] \quad (3)$$

where A_H is the set of axioms that are used in H, $\rho(L)$ is the function that returns the literal corresponding to the predicate of the first-order literal L (e.g., $\rho(john(x)) = john$), and $\delta(A)$ is the *distance function*, which returns the heuristic distance between the antecedents of the axiom A and the conclusions of A. For example, given the knowledge base in Table I and the distance function $\delta(A) = 1$, the value of $hed(p_7(x), p_1(x))$ is $\delta(a_5) + \delta(a_1) = 2$.

In this paper, we define the distance function as $\delta(A) = 1$, for simplicity. In practice, it is necessary to select a proper distance function because the precision of the H.E.D.s depends on the definition of the distance function. For example, in cost-based abduction [11], the distance function better conforms to the evaluation function when using the cost assigned to each axiom for $\delta(A)$.

Since the H.E.D.s depend only on the knowledge base, we

can estimate these in advance. The computational cost of the estimation is $O(N_{pred}^2)$, where N_{pred} is the number of different predicates in the knowledge base.

C. Potential Elemental Hypotheses Creation with A* Search

In this section, we propose an algorithm that efficiently creates the potential elemental hypotheses. We apply an A* search to generate the potential elemental hypotheses and then trim without loss any that are included in the solution hypothesis. Although we employ the same evaluation function as used by *weighted abduction*, our method can be applied to other frameworks which have the properties discussed in Section III.A.

Now, our goal is to efficiently hypothesize the literals that can be combined. Since we cannot know exactly which axiom we should use in order to hypothesize those literals, we search for them by using the H.E.D.s, as follows.

First, set positive values for $dist_{\max}$ and $depth_{\max}$, which are hyperparameters that control the size of the search space and initialize the open set to be an empty set. We denote the distance of the path from a literal p to a literal q as $d(p,q)$ and the estimated distance between p and q as $d^*(p,q)$. We use the distance function $hed(p,q)$ as the heuristic function that provides $d^*(p,q)$. In each step, the following operations are performed:

1) Select the target literal $\hat{q}$, which is expected to result in the least expensive unification with the literals in the open set.
2) Pop $\hat{q}$ off the open set. Enumerate the axioms whose descendant equals $\hat{q}$, and perform backward chaining with each of the axioms with the condition that at least one pair of a literal p_i in the antecedents of the axiom and a literal o in the observations satisfies the following conditions: (i) p_i is considered to be reachable by o (i.e., $hed(p_i, o) \leq dist_{\max}$); (ii) there is no possibility of unification between one of the descendants of p_i and one of the antecedents of o.
3) If a literal in X and one in the potential elemental hypotheses are unifiable, insert the elemental hypotheses of equality between the terms resulting from the unification.

The search is over when the open set is empty.

For example, given the knowledge base shown in Table I and an observation $O = \{p_2(a) \wedge p_6(b,c) \wedge p_7(d)\}$, the first step of the search is performed as shown in Fig. 4; the edges drawn with a solid line represent backward chaining, and those drawn with a dotted line are unifications. The numbers in the balloons connected to the nodes in the open set indicate the estimated distance. In the initial step, since the shortest path is expected to be the one between $p_7(d)$ and $p_2(a)$, the literals $p_2(d)$ and $p_5(u_1)$ are inserted into the open set as the results of backward chainings.

The procedure is shown in Algorithm 1, X is the open set for the search. Each element $x \in X$ is a candidate for the search and has three possible designations: $x.s$ is the start node, $x.c$ is the current node, and $x.g$ is the goal node. The function *isExplanationOf*(x,y) is the binary function that indicates which the literal x explains the literal y (i.e., if x is an antecedent of y), and the function *depth*(p) returns the number of backward chainings that are needed to hypothesize the literal p from the observations.

Algorithm 1: A* search-based potential elemental hypothesis creation

Input: $B, O = \{o_1 \wedge o_2 \wedge \ldots \wedge o_n\}, dist_{\max}, depth_{\max}$
$X \leftarrow \emptyset$
$P \leftarrow \emptyset$
for $i = 1$ to n **do**
  for $j = 1$ to $i - 1$ **do**
    $U \leftarrow getEqualityAssumption(o_i, o_j)$
    $P \leftarrow P \cup U$
    if $hed(o_i, o_j) > 0$ **then**
      $X \leftarrow X \cup x, x.c = o_i \wedge x.s = o_i \wedge x.g = o_j$
      $X \leftarrow X \cup x, x.c = o_j \wedge x.s = o_j \wedge x.g = o_i$
    end if
  end for
end for
while $X \neq \emptyset$ **do**
  $\hat{x} \leftarrow \arg\min_{x \in X} \{d(x.s, x.c) + hed(x.c, x.g)\}$
  for all $a = \{p_1 \wedge \ldots \wedge p_n \Rightarrow q\}$ in B **do**
    $R \leftarrow doBackwardChaining(\hat{x}.c, a)$
    $P \leftarrow P \cup R$
    for all r in R **do**
      for all x in $\{x | x \in X \wedge x.c = \hat{x}.c\}$ **do**
        if $d(x.s, x.c) + hed(x.c, x.g) + \delta(a) \leq dist_{\max} \wedge depth(x.c) < depth_{\max}$ **then**
          $X \leftarrow X \cup \{y | y.s = x.s \wedge y.c = r \wedge y.g = x.g \wedge d(y.s, y.c) = d(x.s, x.c) + \delta(a)\}$
        end if
      end for
      for all p in $P \setminus r$ **do**
        $U \leftarrow getEqualityAssumption(r, p)$
        $P \leftarrow P \cup U$
        if $U \neq \emptyset$ **then**
          $X \leftarrow X \setminus \{x | x.c = r \wedge isExplanation(p, x.g)\}$
          $X \leftarrow X \setminus \{x | x.c = p \wedge isExplanation(r, x.g)\}$
        end if
      end for
    end for
  end for
  $X \leftarrow X \setminus \{x | x.c = \hat{x}.c\}$
end while
return P

Algorithm 2: *doBackwardChaining*(l,a)

Input: $l, a = \{p_1 \wedge p_2 \wedge \ldots \wedge p_n \Rightarrow q\}$
$P \leftarrow \emptyset$
if $\exists \theta, l\theta = q$ **then**
  for $v \in notSubstitutedVars(\{p_1, \ldots p_n\}, \theta)$ **do**
    $\theta \leftarrow \theta \cup \{v/u_i\}; i \leftarrow i + 1$
  end for
  $P \leftarrow P \cup \{p_1, p_2, \ldots, p_n\}\theta$
end if
return P

Algorithm 3: *getEqualityAssumption*(p_1,p_2)

Input: p_1, p_2
$P \leftarrow \emptyset$
if $\exists \theta, p_1\theta = p_2$ **then**
  for all x / y in θ **do**
    $\theta \leftarrow P \cup \{x = y\}$
  end for
end if
return P

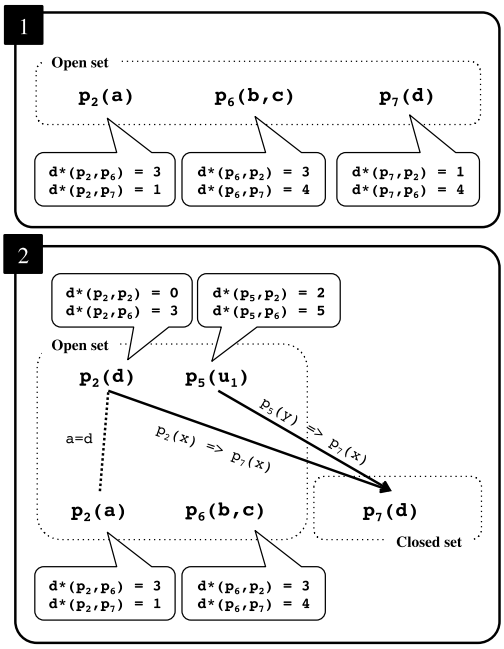


Fig. 4. An example of the creation of potential elemental hypotheses based on an A* search.

Next, we summarize the advantages of this algorithm. First, since this algorithm does not add literals that cannot be included in the solution hypothesis to the potential elemental hypotheses, it can reduce the size of the search space. We believe that this may lead to a more efficient optimization.

Second, this algorithm prevents redundant unifications. For example, given the knowledge base shown in Table I and the observation $O = \{p_7(a) \wedge p_7(b)\}$, let us consider how to generate the potential elemental hypotheses *P*. In Inoue and Inui's algorithm [10], [11], the potential elemental hypotheses generated are $P = \{p_2(a) \wedge p_2(b) \wedge p_1(a) \wedge p_1(b)\}$. However, according to Section III.A, the evaluation of the candidate hypothesis $H = (a = b)$ must be better than the evaluation of $H = \{p_2(a) \wedge p_2(b) \wedge (a = b)\}$ or $H = \{p_1(a) \wedge p_1(b) \wedge p_2(a) \wedge p_2(b) \wedge (a = b)\}$. We have no need to consider backward chainings from observations in this case. Our algorithm can deal with such a heuristic.

Third, this algorithm adds elemental hypotheses to the potential ones in the order of their probability of being included in the solution. Therefore, if the generation of potential elemental hypotheses is interrupted due to a time out, etc., a better suboptimal solution is provided. This property is expected to be much more useful in practice.

D. Parallelization

In the domain of the efficiency of other frameworks for inference, some researchers have adopted the approach of parallelizing the inference by splitting the input into independent subproblems [16]-[19]. We explore a similar method to parallelize abductive inference by using H.E.D.s, which were proposed in the previous section.

First, we consider the condition that two subproblems o_i and o_j are independent. This condition is defined by the particular evaluation function that is used. For instance, in weighted abduction, the conditions can be defined as follows:

1) There is no elemental hypothesis that explains both the literals $p \in o_i$ and $q \in o_j$ (i.e. $\min_{\{(p,q)|p \in o_i \wedge q \in o_j\}} hed(p,q) = \infty$).
2) Equalities between any two terms cannot be hypothesized from o_i and o_j together. In other words, o_i and o_j can share no more than one logical variable.

Given observations *O*, the inference is parallelized via the following process:

1) Split the observations *O* into independent subproblems $\{o_1, o_2, ..., o_n\}$.
2) Compute in parallel the solution hypothesis for each subproblem.
3) Merge the solution hypotheses of the subproblems, and then output the solution hypothesis of *O*.

As mentioned, the computational cost of abduction grows exponentially with the number of observations. Therefore, dividing the observations into subproblems not only reaps the benefits of parallel computing, but it is also expected to reduce the total computational cost.

IV. Experiments

A. Dataset

We used the same dataset as the one used by Inoue and Inui [11]; it consists of sets of observations and a knowledge base. The observation sets were created by converting the development dataset of RTE-2 [1], the task of Textual Entailment Recognition, with the Boxer semantic parser[2]; it consists of 777 observation sets. The average number of literals in each observation set was 29.6.

The knowledge base consists of 289,655 axioms that were extracted from WordNet [20], and 7,558 that were extracted from FrameNet [21]. The number of different predicates in this knowledge base is 269,725.

B. Dataset

For this dataset, we compared the solving times when using our models and when using that of Inoue and Inui (2012), which is currently the state of the art. We will denote their model as *Baseline* and ours as *A*-single* and *A*-parallel*. *A*-based* will be used to refer to both of A*-single and A*-parallel. We also compared the computational costs for pre-estimating the H.E.D.s with various $dist_{\max}$.

In the experiment, the parameter $depth_{\max}$ was 3, and the parameter $dist_{\max}$ of A*-based was 6. We employed weighted abduction [1] as the evaluation function. We defined the distance function $\delta(a) = 1$ for simplicity, and so that the search space on A*-based was equal to that of Baseline.

For our experiments, we used 8-Core Opteron 6174 (2.2 GHz) 128 GB RAM machines. We used a Gurobi optimizer[3], which is a broadly used efficient ILP solver. It is a commercial product but is freely available with an academic license. For *Baseline* and *A*-single*, we ran the whole system on one 8-Core machine, where Gurobi worked in the parallel mode. For A*-parallel, we automatically dispatched the generated parallel subproblems into 4 sets of 8-Core machines.

C. Results

The results of the first experiment are shown in Table II. Here, we excluded from the results those problems for which

[1] http://pascallin.ecs.soton.ac.uk/Challenges/RTE2/
[2] http://svn.ask.it.usyd.edu.au/trac/candc
[3] http://www.gurobi.com/

the whole abductive reasoning took more than 120 seconds in at least one the settings of Baseline, A*-single and A*-parallel; as a result, 707 problems remained out of the original 777 problems. The row **# of literals** shows the average number of literals in the potential elemental hypotheses, the row **# of chains** shows the average number of backward chainings in the potential elemental hypotheses, and the row **# of unifications** shows the average number of unifications in the potential elemental hypotheses.

TABLE II: THE RESULT OF THE COMPARISON BETWEEN OUR METHOD AND THE BASELINE

	Baseline	A*-single	A*-parallel
# of literals	1120	349	349
# of chains	1027	302	302
# of unifications	460	166	166
Time (P-Gen)	0.14	0.13	0.22
Time (Conv)	0.21	0.07	0.07
Time (Solve)	5.93	1.46	0.82
Time (All)	6.29	1.67	1.13
# of timeout	70	33	29

TABLE III: THE COMPUTATIONAL COST OF PRE-ESTIMATING THE H.E.D.S

	Time	File Size
$dist_{max} = 4$	106	0.8GB
$dist_{max} = 6$	1514	5.8GB
$dist_{max} = 8$	7841	28GB

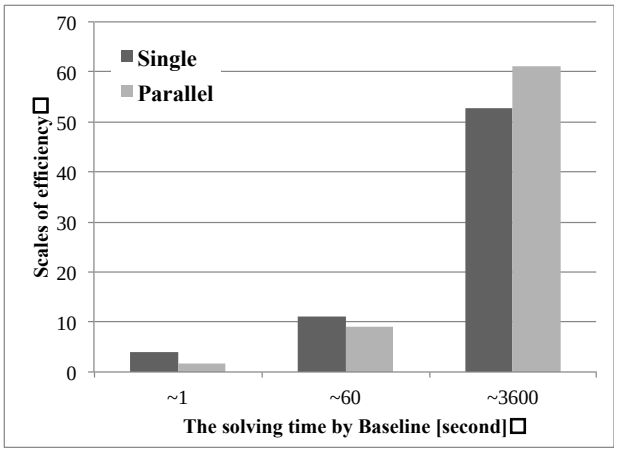


Fig. 5. The comparison between the single thread system and the parallel thread system.

Time (P-Gen) shows the average time (seconds) required to generate the elemental hypotheses, **Time (Conv)** shows the average time (seconds) required to convert the elemental hypotheses into an ILP problem, **Time (Solve)** shows the average time (seconds) required to optimize the ILP problem, and **# of timeout** shows the number of problems that timed out.

From Table II, it can be observed that the number of the generated potential elemental hypotheses is significantly smaller in the proposed A* search-based settings than the baseline, and as a result, the time for reasoning was considerably reduced as indicated in **Time (All)**.

The gain of the efficiency in A*-single and A*-parallel is explored more closely in Fig. 5. Here, the problems are divided into three bins according to the time consumed by the Baseline system. For each bin, the figure shows the gain of the time efficiency of A*-single and A*-parallel compared with Baseline (i.e. the Baseline's inference time (i.e. **Time (All)**) divided by our systems' inference time).

From this figure, it can be onserved that both A*-single and A*-parallel work drastically efficiently compared with the baseline particularly for complex problems. The gain of efficiency by A*-single compared with A*-parallel is not as impressive as expected in the present experimental setting. A*-parallel tended to improve the time efficiency compared with A*-single at least for complex problems. Obviously, however it is arguable in the present setting that the parallelization is worthwhile at the cost of the additional computational resources. We need to further explore the potential of this direction of research.

The costs for pre-estimating the H.E.D.s are compared in Table III. We see that the computational cost and the size of the database increase sharply as $dist_{\max}$ increases. However, in practice, it is sufficient if $dist_{\max}$ is in the range of 4 to 8, and so it is unlikely that this cost can be a bottleneck.

V. CONCLUSION

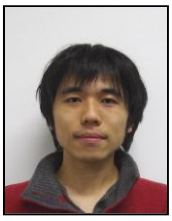
While abduction has long been considered to be a promising framework for making explicit the implicit information in sentences, its computational complexity has hindered the application of abduction to practical NLP problems. In this paper, we proposed a method that makes a significant improvement over the method of Inoue and Inui (2012), which provided the current state-of-the-art system. Specifically, our method is designed to generates as a small number of potential elemental hypotheses as possible by discarding literals which have no chance to constitute the solution (i.e. optimal) hypothesis in an A* search-based fashion. We then conducted an experiment with a considerably large-scale knowledge base for a real-life NLP task. The results shows that the proposed method drastically reduces the size of the search space and significantly improves the computational efficiency of first-order abductive reasoning compared with the state-of-the-art system. The gain tended to be more drastic particularly in complex problems. We also explored a method for parallelizing a given problem by seeking independent subproblems. Our experiment shows that the parallelization tended to improve the time efficiency compared with A*-single at least for complex problems. However, the results also revealed the necessity of further exploration for this direction of research.

In our future work, since our methods have a strong dependence on the precision of the pre-estimates, we will refine the definition of the H.E.D.s. We note that currently the estimation is imprecise when a predicate does not have a concrete meaning and tends to occur with other literals in axioms; for example, this happens with the literals for functional verbs. This problem occurs because an axiom $p_1 \wedge p_2 \Longrightarrow q$ in the knowledge base is split into the axioms $p_1 \Longrightarrow q$ and $p_2 \Longrightarrow q$ during the pre-estimation. Therefore, it is important to determine how to enrich the functionality of the pre-estimation without causing the computational cost to explode.

REFERENCES

[1] J. R. Hobbs, M. Stickel, P. Martin and D. Edwards, "Interpretation as abduction," *Artificial Intelligence*, vol. 63, pp. 69-142, 1993.
[2] N. Inoue, E. Ovchinnikova, K. Inui and J. R. Hobbs, "Coreference resolution with ILP-based weighted abduction," in *Proc. the 24th*

International Conference on Computational Linguistics, December 2012, pp. 1291-1308.
[3] H. T. Ng and R. J. Mooney, "Abductive plan recognition and diagnosis: a comprehensive empirical evaluation," in *Proc. the Third International Conference on Principles of Knowledge Representation and Reasoning*, 1992, pp. 499-508.
[4] 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 *Proc. 2011 IWCS*, 2011, pp. 225-234.
[5] R. Raina, A. Y. Ng, and C. D. Manning, "Robust textual inference via learning and abductive reasoning," in *Proc. AAAI-2005*, 2005.
[6] P. Singla and P. Domingos, "Abductive Markov logic for plan recognition," in *Proc. AAAI-2011*, 2011, 1069-1075.
[7] N. Chambers and D. Jurafsky, "Unsupervised learning of narrative schemas and their participants," in *Proc. the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, 2009, pp. 602–610.
[8] S. Schoenmackers, J. Davis, O. Etzioni, and D. Weld, "Learning first-order horn clauses from web text," in *Proc. EMNLP-2010*, 2010, 1088–1098.
[9] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: A collaboratively created graph database for structuring human knowledge," in *Proc. the 2008 ACM SIGMOD International Conference on Management of Data*, 2008, New York, NY, USA: ACM, pp. 1247-1250.
[10] N. Inoue and K. Inui, 2011, "ILP-Based reasoning for weighted abduction," in *Proc. AAAI Workshop on Plan, Activity and Intent Recognition*.
[11] N. Inoue and K. Inui, "Large-scale cost-based abduction in full-fledged first-order predicate logic with cutting plane inference," in *Proc. the 13th European Conference on Logics in Artificial Intelligence*, September 2012, pp. 281-293.
[12] E. Charniak and R. P. Goldman, "A probabilistic model of plan recognition," in *Proc. AAAI-91*, 1991, pp. 160-165.
[13] S. Raghavan and R. J. Mooney, "Bayesian abductive logic programs," in *Proc. STARAI-2010*, 2010, pp. 82-87.
[14] J. Blythe, J. R. Hobbs, P. Domingos, R. J. Kate, and R. J. Mooney, "Implementing Weighted Abduction in Markov Logic," in *Proc. IWCS*, 2011, pp. 55-64.
[15] M. Richardson and P. Domingos, "Markov Logic Networks," *Machine Learning*, pp. 107-136, 2006.
[16] V. Jojic, S. Gould and D. Koller, "Accelerated Dual Decomposition for MAP Inference," in *Proc. the 25th International Conference on Machine Learning*, 2010, pp. 503-510.
[17] J. E. Gonzalez, Y. Low, C. E. Guestrin and D. O'Hallaron, "Distributed parallel inference on large factor graphs," in *Proc. the 25th Conference on Uncertainty in Artificial Intelligence*, 2009.
[18] F. Niu, C. Zhang, C. Re´ and J. W. Shavlik, "Scaling Inference for Markov Logic via dual decomposition," in *Proc. the IEEE International Conference on Data Mining*, 2012, pp. 1032-1037.
[19] J. Urbani, S. Kotoulas, E. Oren, and F. Van Harmelen, "Scalable distributed reasoning using mapreduce," in *Proc. the 8th International Semantic Web Conference*, 2009, pp. 634-649. Springer.
[20] C. Fellbaum, *WordNet: An Electronic Lexical Database*, MIT Press, 1998.
[21] J. Ruppenhofer, M. Ellsworth, M. Petruck, C. Johnson, and J. Scheffczyk, 2010, "FrameNet II: Extended theory and practice," Technical report, Berkeley, USA.



Kazeto Yamamoto received the B.E degree of engineering from Tohoku University, in 2011 and the M.S. degree in information science from Tohoku University in 2013. He is currently a Ph.D. student in Tohoku University. His research interests include natural language processing, logical inference and machine learning.

Naoya Inoue received his M.S. degree of engineering from Nara Institute of Science and Technology in 2010 and his Ph.D. degree in information science from Tohoku University in 2013. He is currently a post-doc researcher at Tohoku University and works as a researcher for DENSO Research Laboratories. His research interests are in inference-based discourse processing and language grounding problems.

Kentaro Inui received his doctorate degree of engineering from Tokyo Institute of Technology in 1995. He has experience as an assistant professor at Tokyo Institute of Technology and an associate professor at Kyushu Institute of Technology and Nara Institute of Science and Technology, he has been a professor of Graduate School of Information Sciences at Tohoku University since 2010. His research interests include natural language understanding and knowledge processing. He currently serves as the IPSJ director and ANLP director.

Yuki Arase received her B.E., M.I.S., and Ph.D. of information science in 2006, 2007, and 2010 respectively from Osaka University, Japan. She joined Microsoft Research in Beijing as an associate researcher in April 2010. In her Ph.D study, she has studied HCI on mobile devices, especially how to present a large web page on a small screen. She also worked on web data mining. In Microsoft Research, she has started working on English/Japanese natural language processing. Currently, she is an associate professor at the graduate school of information science, Osaka University. Her current research interests include English paraphrase detection, Japanese-English statistical machine translation, and Web data mining.

Jun'ichi Tsujii received his B.E, M.E. and Ph.D. degrees in electrical engineering from Kyoto University, Japan, in 1971, 1973, and 1978, respectively. He was an assistant professor and associate professor at Kyoto University before taking up the position of professor of computational linguistics at the University of Manchester's Institute for Science and Technology (UMIST) in 1988. Since 1995, he has been a professor of Department of Computer Science, the University of Tokyo. He is also a professor of text mining at the University of Manchester (part-time) and Research Director of the UK National Centre for Text Mining (NaCTeM) since 2004. He was the president of the Association for Computational Linguistics (ACL) in 2006 and has been a permanent member of the International Committee on Computational Linguistics (ICCL) since 1992.