

A*探索に基づく仮説推論の効率化

山本 風人^{1,a)} 井之上 直也^{1,b)} 乾 健太郎^{1,c)} 荒瀬 由紀^{2,d)} 辻井 潤一^{2,e)}

概要: 仮説推論は、観測に対する最良の説明を求める推論である。仮説推論は、文章に明示されていない情報の顕在化を行うための有望な枠組みとして考えられてきた一方で、最良の説明を求めるための計算時間が入力規模に対して指数関数的に増加するため、実問題への適用事例は未だ限定的であるのが現状である。そこで我々は、仮説推論の探索空間を効率良く削減するために、A*探索に基づく説明候補生成アルゴリズムを提案する。より具体的には、複数の証拠により支持されない仮説リテラルを、効率良く探索空間から排除する手法を提案する。また、評価実験を通して、最新の仮説推論アルゴリズムの計算効率が提案手法により大幅に改善されることを報告する。

キーワード: 仮説推論、グラフ探索、A*探索

Efficient Estimation of Plausible Abductive Hypotheses Using A* Search

KAZETO YAMAMOTO^{1,a)} NAOYA INOUE^{1,b)} KENTARO INUI^{1,c)} YUKI ARASE^{2,d)} JUNICHI TSUJII^{2,e)}

Abstract: Abduction is inference to the best explanation. While abduction has long been considered as a promising framework for natural language processing (NLP), its computational complexity hinders applying abduction to practical NLP problems. In this paper, we adapt an A* search-based technique for abduction in order to prune its search space efficiently. More specifically, we present an algorithm to effectively eliminate literals which do not introduce any unification from the search space. Our experiment reveals that our search-based technique considerably reduces the size of the search space and improves the efficiency of the state-of-the-art inference algorithm for abduction on first-order logic.

Keywords: Abduction, Graph search, A* search

1. はじめに

文章に明示されていない情報(照応関係、登場人物の意図など)を顕在化する処理(Making the implicit explicit)は、情報抽出やQAなどの多くの自然言語処理のタスクにおいて重要な技術である。我々は、このような情報の顕在化を行う枠組みとして、仮説推論を用いた言語理解の枠組みの構築に取り組んでいる。

仮説推論とは、与えられた観測に対して、評価関数を最

大化する最良の説明を求める推論である。仮説推論を言語理解に適用する試みは1980年代から1990年代に多く見られたが、そのうち最も代表的な枠組みがHobbsら[4]のInterpretation as Abduction (IA)である。Hobbsらは、照応解析や談話関係などの談話理解に関する種々のタスクを、仮説推論(アブダクション)のタスクとして自然に定式化できることを示した。たとえば、“John went to the bank. He got a loan.”という文章に対して解釈を行う場合を考えてみよう。IAでは、世界知識を背景知識、解釈の対象となる文章を観測として仮説推論を行い、最良の説明を解釈の結果として出力する。図1に、仮説推論による文の解釈の様子を示す。解釈の結果から、「JohnとHeが共参照関係である」「went to the bankの目的はgot a loanであった」といった、文章に明示されていない情報が顕在化されていること

¹ 東北大学

² Microsoft Research Asia

^{a)} kazeto@ecei.tohoku.ac.jp

^{b)} naoya-i@ecei.tohoku.ac.jp

^{c)} inui@ecei.tohoku.ac.jp

^{d)} yukiar@live.com

^{e)} jtsujii@microsoft.com

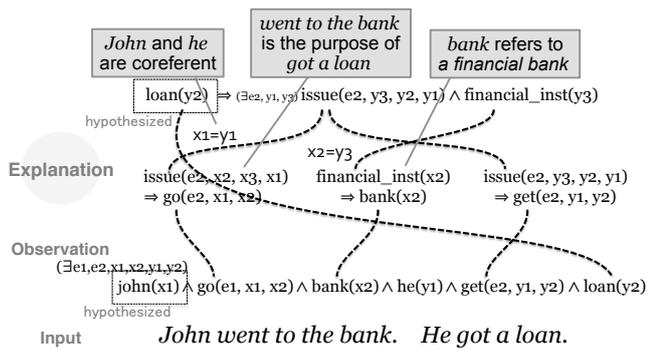


図1 仮説推論による言語理解の例

がわかる。

仮説推論による言語理解のモデル化の利点としては、各サブタスク間の相互依存関係を考慮し、それぞれのタスクにおいて一貫した解釈のうち、最も良い解釈を出力するという処理を自然にモデル化できるという点がある。情報の顕在化における最新の研究プロジェクトである Machine Reading におけるいくつかの推論モデルも、仮説推論を用いて自然にモデル化できると考えられる [10]。

IA の枠組みは、1990 年代-2000 年代を経てふたたび注目を浴び始めている [1], [9], [16], [17]。これには、大きく 2 つの背景があると考えられる。ひとつめに、仮説推論を実用規模の問題に適用できる程度の、大規模な世界知識を取り揃える技術が昔に比べて大幅に成熟してきたことが挙げられる [2], [3], [13], [14]。例えば文献 [9] では、WordNet [3] と FrameNet [13] を用いて約数十万の推論規則からなる背景知識を構築し、含意関係認識のタスクに IA を適用している。ふたつめの背景には、効率的な仮説推論エンジンが提案されたことが挙げられる [6]。例えば文献 [17] では、約数十万の推論規則からなる背景知識を用いて含意関係認識のデータセットに対して推論を行い、先行研究より大幅に高速な推論を行えたことが報告されている。

しかしながら、仮説推論の計算時間は、入力となる背景知識や観測の規模に対して指数関数的に増加するため、先行研究の高速化は必ずしも十分であるとはいえない。例えば、文献 [8] では、仮説推論を共参照解析の問題に適用しているが、背景知識のカバレッジの低さが性能向上を妨げる原因となっており、仮説推論の背景知識を更に大規模化する必要があることを示唆している。

そこで本稿では、仮説推論における説明候補の生成処理を探索問題として捉え、A*探索に基づく探索手法を適用することによって、効率的に推論を行う手法を提案する。より具体的には、複数の証拠により支持されない仮説リテラルを、効率良く探索空間から排除する手法を提案する。なお、本稿では、仮説の評価関数として重み付き仮説推論の評価関数を用いているが、他の仮説推論の枠組みに対しても容易に拡張することができる。

本稿では、まず重み付き仮説推論と、ILP による仮説推論の定式化について説明したあと (2 節)、A*探索に基づく推論効率化の手法について説明する (3 節)。次に、提案手法と既存手法との比較実験の結果について報告し (4 節)、最後に今後の展望を述べる (5 節)。

2. 先行研究

一階述語論理の仮説推論の高速化に対する最近の取り組みとして、Markov Logic Networks (MLNs) [12] や Bayesian Logic Programs [11] などの演繹推論の枠組みの上で、仮説推論をエミュレートする研究がある [11][1][15]。例えば、文献 [15] では、仮説推論の背景知識 (e.g., $\{P_1 \Rightarrow Q, P_2 \Rightarrow Q, \dots, P_n \Rightarrow Q\}$) の逆向き含意と、説明の排他性制約を表す論理式の集合 (e.g., $\{Q \Rightarrow P_1, Q \Rightarrow P_2, \dots, Q \Rightarrow P_n, P_1 \Rightarrow \neg P_2, P_2 \Rightarrow \neg P_3, \dots, P_{n-1} \Rightarrow P_n\}$) を MLNs の背景知識として利用することにより、仮説推論を実現する手法を示している。これらの手法は、機械学習分野で培われた高速な推論アルゴリズムを容易に利用可能であるが、大規模問題への適用に耐えうる性能には至っていないことが報告されている [1]。これらの研究における速度向上の妨げの主要因としては、一階述語論理で書かれた背景知識・観測を、命題論理に変換する過程 (*grounding*) で、背景知識の規模が爆発的に増加してしまうことが挙げられている。

この問題に対処するために、井之上 [6] は、背景知識・観測を命題論理に変換せずに推論する手法 (*lifted inference*) を提案し、劇的な高速化を達成した。また、一階述語論理上の仮説推論問題を整数線形計画問題 (ILP) として定式化し、ILP ソルバの高速な求解アルゴリズムを用いて更なる高速化を達成した [6][5]。この手法は、MLNs に基づく手法 [1] と比較して、約 500 倍の高速化が実現できていることが報告されており、現状で最も高速な手法であるといえる。しかしながら、大規模な背景知識を用いた場合には、ILP 問題の規模 (変数・制約の数) が大きくなってしまい、推論速度が著しく低下するという問題がある。本研究は、井之上 [6] の手法の問題点を改善するために、不必要な仮説リテラルを探索空間から事前に排除することで、ILP 問題の規模をできるだけ小さく抑え、推論の高速化を狙うものである。

3. 背景

3.1 重み付き仮説推論

仮説推論とは、与えられた観測に対して最良の説明を求める推論である。形式的には次のように定義される。

Given: 背景知識 B 、観測 O 。ただし B, O は一階述語論理式の集合。

Find: 仮説 H 。ただし H は $H \cup B \models O, H \cup B \not\models \perp$ を満たす一階述語論理式の集合。

本稿では、上の条件を満たす H を候補仮説と呼び、そのな

かで最良の仮説を解仮説と呼ぶこととする。候補仮説の中から解仮説を選択する際の評価方法は様々な手法が存在するが、我々は重み付き仮説推論の枠組み [4] を用いる。重み付き仮説推論では、観測 O と仮説 H は存在限量されたリテラルの連言として表現され、各リテラルは正の実数のコストを持つ。直感的には、リテラルのコストとはそのリテラルの不確かさであり、コストが高いリテラルを説明するためにはより多くの根拠が必要となる。また、背景知識 B はホーン節: $P_1^{w_1} \wedge P_2^{w_2} \wedge \dots \wedge P_n^{w_n} \Rightarrow Q$ の形式を持つ一階述語論理式 (公理) の集合である。前件のリテラルには正の実数の重みが割り当てられている。直感的には、背景知識の重みはその背景知識による推論のこじつけ度合いを表し、重みが高い背景知識によって仮説されたリテラルは高いコストを持つこととなる。

重み付き仮説推論では、以下の二つの操作を観測に対して逐次適用することで、候補仮説を生成する:

(i) 後ろ向き推論は、背景知識 $P_1(x)^{w_1} \wedge P_2(x)^{w_2} \wedge \dots \wedge P_n(x)^{w_n} \Rightarrow Q(x) \in B$ と、観測または仮説リテラル $Q(a)$ を前提として、新しいリテラルの集合 $\{P_i(a)\}_{i=1}^n$ を導き、候補仮説に追加する処理である。このとき、生成されたリテラル $P_i(a)$ のコスト $c_{P_i(a)}$ を、リテラル $Q(a)$ のコスト $c_{Q(a)}$ と、背景知識の重み w_i の掛け算により計算し ($c_{P_i(a)} = w_i \cdot c_{Q(a)}$)、リテラル $Q(a)$ のコストを 0 とする。例えば、 $O = \exists x(q(x)^{S10})$ 、 $B = \forall x(p(x)^{1.2} \Rightarrow q(x))$ が与えられた場合は、 $H_1 = \exists x(q(x)^{S10})$ 、 $H_2 = \exists x(q(x)^{S0} \wedge p(x)^{S12})$ の二つの候補仮説を生成する。(ii) 単一化は、同一の述語を持つ 2 つのリテラルを合成し、互いの項が等価であるという仮説を置く操作である。この際、2 つのリテラルを持つコストのうち大きい方のコストを 0 にする。例えば、 $O = \exists x, y(p(x)^{S10} \wedge p(y)^{S20} \wedge q(y)^{S10})$ が与えられた場合は、 $H = \exists x, y(p(x)^{S10} \wedge q(y)^{S10} \wedge x = y)$ を候補仮説として生成する。

背景知識 B と観測 O に対する解仮説 $\hat{H}_{B,O}$ は、候補仮説の集合 \mathbb{H} の中で最小のコストを持つ候補仮説、すなわち最も蓋然的な仮説と定義される。

$$\hat{H}_{B,O} = \arg \min_{H \in \mathbb{H}} Cost_{B,O}(H) \quad (1)$$

ここで、 $Cost_{B,O}(H)$ は候補仮説 H が持つコストを返す関数である。候補仮説のコストは、候補仮説に含まれるリテラル (要素仮説) のコストの和として定義され、次の式により表現できる。

$$Cost_{B,O}(H) = \sum_{o \in O} c_o + \sum_{p \in H} c_p \quad (2)$$

$$= \sum_{o \in O} c_o + \sum_{p \in H} \left[\prod_{w \in Chain(p)} w \right] \hat{c}_{Obs(p)} \quad (3)$$

ここで、 c_p はリテラル p が持つコスト、 \hat{c}_p はリテラル p が後ろ向き推論や単一化によってコストが 0 になる前に持つ

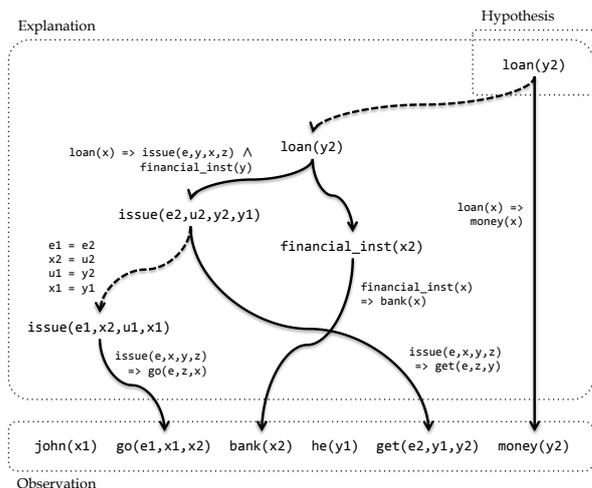


図 2 非循環有向グラフによる仮説の表現

ていたコストを表す。Chain(p) はリテラル p を仮説するために用いられた背景知識の重みの集合、Obs(p) はリテラル p が説明している観測を表す。背景知識はホーン節であるので、 p から Obs(p) へ辿る道筋は枝分かれせずに一つに定まる。

重み付き仮説推論の仮説評価指標を用いる利点としては、仮説の特殊性を評価できることが挙げられる。仮説推論では、後ろ向き推論が適用されるごとに仮説の特殊性は高くなるが、一般には仮説が特殊になるほど仮説を支持する証拠は少なくなるため、信頼度が高い範囲で推論を停止できることが望ましい。重み付き仮説推論では、仮説コストの伝播と単一化の操作によって、これが自然に実現できているといえる。

仮説推論においては、図 2 のように非循環有向グラフを用いて、仮説と観測間の関係を表現することができる。リテラルおよび変数間の等価性はグラフのノードとして表現され、逆向き推論や単一化などの操作はグラフ上ではエッジとして表現される。観測 O と仮説 H によって定義されるこのようなグラフを $G_{O,H}$ とすると、 (O,H) と $G_{O,H}$ の間の変換は互いに全単射性が成り立ち、 $G(O,H_1) = G(O,H_2)$ は $H_1 = H_2$ に対する必要十分条件となる。

3.2 整数線形計画問題による仮説推論の定式化

仮説推論における最適解の導出は、組み合わせ最適化問題の一つであり、その計算量は観測や背景知識の規模に対して指数的に増大する。そのため、計算速度問題は仮説推論の実用において重大な問題であり続けている。この問題に対して井之上らは、重み付き仮説推論における最適解の導出を整数線形計画問題 (Integer Linear Programming, ILP) として定式化することによって、効率的に推論を行う手法を提案した。彼らの手法は、大きく分けて次の 3 つのステップで構成される (詳しくは、文献 [7] を参照されたい)。

Step 1. 観測 O と背景知識 B から導くことのできる、全て

Algorithm 1 潜在仮説集合の生成

```

1:  $P_0 \leftarrow O$ 
2: for  $d = 1$  to  $depth_{max}$  do
3:    $P_d \leftarrow \emptyset$ 
4:   for all  $l \in P_{d-1}$  do
5:     for all  $a = \{p_1 \wedge p_2 \wedge \dots \wedge p_n \Rightarrow q\} \in B$  do do
6:        $P_d \leftarrow P_d \cup getPotentialElementalExplanation(l, a)$ 
7:     end for
8:   end for
9: end for
10:  $P \leftarrow \bigcup_{d=0}^{depth_{max}} P_d$ 
11: for all  $p_1, p_2$  in  $P$  do
12:    $P \leftarrow P \cup getEqualityAssumption(p_1, p_2)$ 
13: end for
14: return  $P$ 

```

Algorithm 2 $getPotentialElementalExplanation(l, a)$

```

Ensure:  $a = \{p_1 \wedge p_2 \wedge \dots \wedge p_n \Rightarrow q\}$ 
1:  $P \leftarrow \emptyset$ 
2: if  $\exists \theta, l\theta = q\theta$  then
3:   for  $v \in notSubstitutedVars(\{p_1, p_2, \dots, p_n\}, \theta)$  do
4:      $\theta \leftarrow \theta \cup \{v/u_i\}; i \leftarrow i + 1$ 
5:   end for
6:    $P \leftarrow P \cup \{p_1, p_2, \dots, p_n\}\theta$ 
7: end if
8: return  $P$ 

```

Algorithm 3 $getEqualityAssumption(p_1, p_2)$

```

1:  $P \leftarrow \emptyset$ 
2: if  $\exists \theta, p_1\theta = p_2\theta$  then
3:   for all  $x/y$  in  $\theta$  do
4:      $P \leftarrow P \cup \{x = y\}$ 
5:   end for
6: end if
7: return  $P$ 

```

の要素仮説の集合 P を生成する（以降、潜在仮説集合と呼ぶ）。 P を生成する具体的な手順は、アルゴリズム 1 に示される。ここで、 $notSubstitutedVars(\{p_1, p_2, \dots, p_n\}, \theta)$ は、リテラルの集合 $\{p_1, p_2, \dots, p_n\}$ に置換 θ を適用した際に、置換されなかった変数の集合である。 $depth_{max}$ はユーザによって任意の自然数をとるパラメータであり、この値によって推論の探索範囲が制御される。この値が小さいほど潜在仮説集合の規模は小さくなる一方、得られる解の精度は低くなる。

Step 2. 潜在仮説集合 P に基づき、最良仮説の探索問題を ILP により定式化する。 B のもとでの O に対する候補仮説は、 P の要素の任意の組合せとして表現できるため、 P の各要素を解仮説に含めるか否かを表す 0-1 ILP 変数を導入する。また、各リテラルが持つコストを、ILP 問題の目的関数における、ILP 変数の係数として設定する。

Step 3. ILP ソルバにより、生成した ILP 問題に対する最適解を求めることで、解仮説を求める。

この手法の推論効率は、**Step. 2** から分かるように、潜在

仮説集合 P の規模に大きく依存する。しかしながら、井之上らが提案した潜在仮説集合の生成処理（アルゴリズム 1）には、解仮説に含まれないことが保証されるリテラル（例えば、単一化を導かないリテラルなど）を P に追加してしまう、という問題点がある。

例えば、 $O = P \wedge Q, B = \{R \Rightarrow P, S \Rightarrow P, S \Rightarrow Q\}$ に対しては、アルゴリズム 1 により $P = \{P, Q, R, S\}$ を得る。しかしながら、 R は他のどのリテラルとも単一化されないため、解仮説 $H_{B,O}$ には含まれない。このようなリテラルを P の中に保持することは、不必要に ILP 問題を複雑化させる一因となる。特に、背景知識の規模が大きい場合には、致命的な問題になると考えられる。

3.3 A*探索アルゴリズム

A*探索アルゴリズムとは、グラフの各ノード n からゴールノード n_g までの最短経路の距離の推定値 $d^*(n, n_g)$ が既知の場合に、任意のノードからゴールノードまでの最短経路を効率的に解くアルゴリズムである。点 x, y の間の最短経路の距離を $d(x, y)$ 、始点となるノードを n_s としたとき、 n_s から n_g までの最短経路は、 $d(n_s, n) + d^*(n, n_g)$ が小さい順に探索していくことで求まる。このとき、最初に n_g に辿り着いた経路が最短経路となる。

ただし、このアルゴリズムで求まる経路が最短経路であることが保証されるためには、最短距離の推定値 $d^*(n, n_g)$ が次の条件式を満たしている必要がある。

$$\forall n, 0 \leq d^*(n, n_g) \leq d(n, n_g) \quad (4)$$

4. A*アルゴリズムに基づく潜在仮説集合の生成

本節では、3.2 節で述べた問題に対処するために、ILP 問題の規模をできるだけ小さくすることを目的として、潜在仮説集合の生成処理（アルゴリズム 1）を改良する。より具体的には、A*探索を用いて逆向き推論の有用性を評価しながら、(1) 解仮説に含まれる要素仮説を排除せずに、(2) 潜在仮説集合の規模を可能な限り小さくすることにより、推論の効率化を目指す。

4.1 効率化の方針

まず、提案手法において用いる効率化の基本方針について述べる。仮説推論の評価関数は、基本的に「より小さな仮説で、より多くの観測を説明できる仮説」に対して高い評価が与えられるよう定義される。従って、重み付き仮説推論を含めた多くの仮説評価関数は、次のような性質を持つ：

- (1) 逆向き推論を行うと、仮説の評価値が低下する
- (2) リテラル間の単一化を行うと、仮説の評価値が上昇する

(3) 仮説の評価値が上昇するのは、リテラル間の単一化を行ったとき、かつそのときに限る

これらの性質から、演繹的に、解仮説に含まれる任意のリテラル L に対する十分条件として、次のいずれかが成立する: (1) L が単一化されている; (2) L は単一化されていないが、 L と同時に仮説された $L_{sibling}$ が単一化されている; (3) L は単一化されていないが、 L を説明する別のリテラル L_{expl} が単一化されている。^{*1}

以上の議論から、潜在仮説集合の生成の際には、単一化を導く見込みがある推論（つまり、推論した仮説が単一化されるか、推論した仮説を介して単一化が起きる場合）を優先的に行うことで、潜在仮説集合の規模を効率的に縮小することができると思われる。本研究では、このアイデアを潜在仮説集合の生成処理に反映するために、A*探索に基づく生成アルゴリズムを提案する。

4.2 A*探索における距離推定値の計算

本節では、A*探索アルゴリズムを利用する上で必要となる、距離の推定値の計算について述べる。

まず、本手法における最短経路のコストを定義する。A*探索において、スタートノード n_s から、あるノード n_x を経由してゴールノード n_g に到達する際の最短経路を考えたとき、この最短経路のコストを $d_{n_x}(n_s, n_g)$ とすると、次のような関係式がなりたつ：

$$d_{n_x}(n_s, n_g) = d(n_s, n_x) + d(n_x, n_g) \quad (5)$$

ここで $d(x, y)$ はノード x からノード y までの最短経路のコストを表す。この関係式はA*探索アルゴリズムにおいて重要な前提となるため、本手法におけるリテラル間の最短経路コストについてもこの条件式が満たされるように定義しなければならない。

そこで本稿では、リテラル x, y の間の最短経路コスト $d(x, y)$ を次のように定義する：

$$d(x, y) = \min_{H \in \{H | H \cup B \models \{x, y\}\}} \sum_{a \in A_H} \delta(a) \quad (6)$$

ここで H はリテラルの集合 $\{x, y\}$ を説明するような仮説、 A_H は仮説 H が $\{x, y\}$ を説明するまでに経由する公理のリスト、 $\delta(a)$ は公理 a を用いた逆向き推論を表すエッジに割り当てられるコストを返すような任意の関数を表す。

次に、 $d(x, y)$ に対する推定値 $d^*(x, y)$ について考える。A*探索を適用するためには、推定値 $d^*(x, y)$ が式4の条件式を満たしている必要がある。そこで本稿では、距離の推定値として、(1) 公理の含意関係と、(2) リテラルの引数を捨象した無向グラフ上での距離を用いる。形式的には、 $d^*(x, y)$ を次のように定義する：

^{*1} 形式的な証明については、今後の課題とする。

$$B^* = \bigcup_{p_1 \wedge \dots \wedge p_n \Rightarrow q \in B} \left[\bigcup_{i=1}^n \rho(p_i) \Leftrightarrow \rho(q) \right] \quad (7)$$

$$d^*(x, y) = \min_{H \in \{H | H \cup B^* \models \{\rho(x), \rho(y)\}\}} \sum_{a \in A_H} \delta(a) \quad (8)$$

ここで、 $\rho(L)$ は一階述語論理で記述されたリテラル L から、述語部分を取り出したリテラルを表す（例えば、 $\rho(p(x, y)) = p$ ）。

リテラル x, y に対する推定値 $d^*(x, y)$ の値を計算する例を図3に示す。なお、 $\delta(a) = 1$ とする。図に示すように、 B^* は無向グラフとして表すことができ、 d^* は無向グラフにおけるノード間の距離とみなすことができる。

なお、任意のリテラル対 (x, y) に対する推定値 $d^*(x, y)$ の値は背景知識 B のみによって決定されるため、全ての述語のペアに対する推定値は前もって計算しておくことができる。また、その際の時間・空間計算量は、背景知識における述語の異なり数を N_B とすると、 $O(N_B^2)$ となる。

4.3 潜在仮説集合の生成

本節では、前節で導出した $d^*(x, y)$ を用いて潜在仮説集合を生成する手順について述べる。

まず最初に、簡単な例として、図3における背景知識と推定値を用いて、観測 $O = \{p_6(a, b), p_7(c)\}$ に対する潜在仮説集合を作ることを考える。ここでこの目的は $p_6(a, b), p_7(c)$ から仮説したリテラルが互いに単一化するような推論パスを効率良く列挙する事、即ち $p_1(a), p_1(c)$ を見つけ出すことである。観測が与えられた時点では、どの公理を用いて逆向き推論を行えばそれらのリテラルが得られるかは自明でないため、前節で求めた $d^*(x, y)$ を手がかりとして探索を行うことになる。このとき、探索は以下の様な方針に基づいて行われる：

- 推論の方向は公理の逆向きに限定されているため、探索は $p_6(a, b), p_7(c)$ の双方から行う必要がある。
- 探索する順序はA*探索に基づいて決定する。すなわち、観測からの距離とゴールへの推定距離の和が小さいものから順に実行していく。
- $p_6(a, b), p_7(c)$ から仮説されたリテラル同士が単一化した時点で探索を終了する。

以上の方針に基づいて探索を行った様子を図4に示す。ここで、実線で囲まれたリテラルはA*探索における確定済みノードを表し、そうでないリテラルは未確定ノードを表す。また実線のエッジは確定済みの推論を表し、破線のエッジは未確定の推論を表す。未確定ノードから出る吹き出し内の数字は、そのノードに関する最短コストの推定値 $d(n_{start}, n) + d^*(n, n_{goal})$ の値を表す。例えば $p_3(a)$ の最短経路コストの推定値は、スタートノードからの移動距離が $d(p_6(a, b), p_3(a)) = 1$ 、ゴールノードまでの推定コストが（図3より） $d^*(p_3(a), p_7(c)) = 2$ であることから

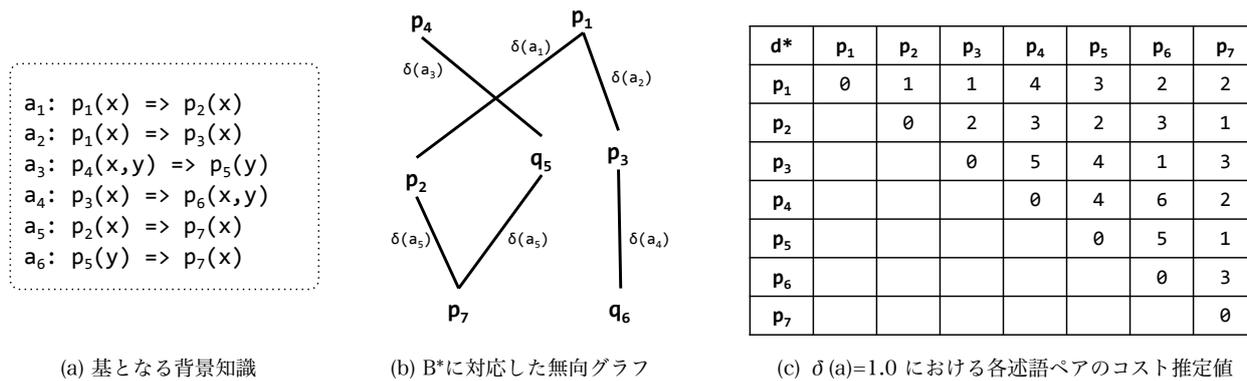


図3 コスト推定値の計算

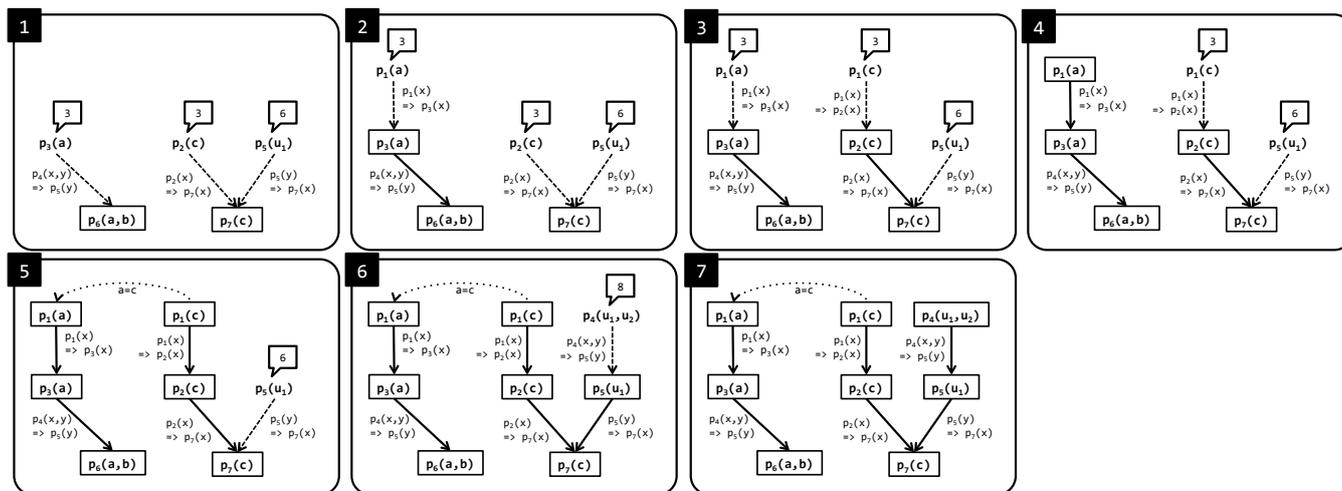


図4 A*探索に基づく潜在仮説集合生成の例

$d(n_{start}, n) + d^*(n, n_{goal}) = 3$ と求められる。この図から、最短コストの推定値が小さい順に推論を進めていくことで、最終的に単一化が起こり得ない $p_5(u_1), p_4(u_1, u_2)$ などの推論よりも先に、 $p_1(a), p_1(c)$ を導くような推論が優先的に行われていることが分かる。

次に、観測がリテラル o_1, o_2, \dots, o_n から成る場合を考える。この時の潜在仮説集合の生成アルゴリズムはアルゴリズム4で示される。ここで、 $dist_{max}$ はユーザによって任意の正值をとるパラメータであり、この値によって推論の探索範囲が制御される。アルゴリズム1における $depth_{max}$ と同様に、この値が小さいほど潜在仮説集合の規模は小さくなる一方、得られる解の精度は低くなる。 X の各要素 x はA*探索における未確定ノードを表し、 x_{start} はスタートノードを、 x_{self} は未確定ノード自身を、 x_{goal} はゴールノードを、 x_{dist} はスタートノードからの移動距離を表す。また、 $isExplanationOf(x, y)$ はリテラル x がリテラル y の説明になっている時、すなわちグラフにおいて y が x の祖先であるような場合に真を返すような二値関数である。

最後に、このような手順で潜在仮説集合を生成することの利点をまとめる。まず、推定値を予め計算しておくことにより、任意の観測リテラルのペアの間を繋げる推論パス

が存在し得るかどうかを即時に判別可能であるという点が挙げられる。すなわち、ある観測リテラルについて他のどの観測リテラルとも単一化する可能性が無い場合、そのリテラルについては評価関数が増加する推論パスが存在しないことから、そのリテラルは推論の対象から除外できる。このことは、潜在仮説集合の規模を効率化する上で重要である。次に、推定距離に近い順に繋げていくという性質により、潜在仮説集合の生成が途中で中断されるような場合に、従来の手法と比べてより良い解が得られる可能性が高くなるという点が挙げられる。大量の観測を処理したい場合など、ある単位時間の中でより良い解を得たい状況において、この特性は非常に有用だと考えられる。

5. 評価実験

本節では、提案手法による推論効率の改善効果を確認するために、既存手法との比較実験を行った結果を報告する。

5.1 データセット

井之上ら [5] と同様のデータセット(観測・背景知識)を用いた。観測は、含意関係認識のタスクである RTE-2 の開

Algorithm 4 A*探索に基づく潜在仮説集合の生成

Input: $B, O = \{o_1, o_2, \dots, o_n\}, dist_{max}$

```

1:  $P \leftarrow \emptyset$ 
2:  $X \leftarrow \emptyset$ 
3: for  $i = 1$  to  $n$  do
4:   for  $j = 1$  to  $i - 1$  do
5:     if  $d^*(o_i, o_j) \leq dist_{max}$  then
6:        $X \leftarrow x, x_{start} = o_i \wedge x_{self} = o_i \wedge x_{goal} = o_j \wedge x_{dist} = 0$ 
7:        $X \leftarrow x, x_{start} = o_j \wedge x_{self} = o_j \wedge x_{goal} = o_i \wedge x_{dist} = 0$ 
8:     end if
9:   end for
10: end for
11: while  $X \neq \emptyset$  do
12:    $x \leftarrow \arg \min_{x \in X} \{x_{dist} + d^*(x_{self}, x_{goal})\}$ 
13:   for all  $a = \{p_1 \wedge p_2 \wedge \dots \wedge p_n \Rightarrow q\}$  in  $B$  do
14:      $R \leftarrow getPotentialElementalExplanation(x_{self}, a)$ 
15:      $P \leftarrow P \cup R$ 
16:     for all  $r$  in  $R$  do
17:        $hasReached \leftarrow false$ 
18:       for all  $p$  in  $P$  do
19:          $U \leftarrow getEqualityAssumption(r, p)$ 
20:          $P \leftarrow P \cup U$ 
21:         if  $U \neq \emptyset \wedge isExplanationOf(r, x_{goal})$  then
22:            $hasReached \leftarrow True$ 
23:            $X \leftarrow X \setminus y, y_{start} = x_{goal} \wedge y_{self} = p \wedge y_{goal} = x_{start}$ 
24:         end if
25:       end for
26:       if  $\neg hasReached \wedge x_{dist} + \delta(a) + d^*(r, x_{goal}) \leq dist_{max}$  then
27:          $X \leftarrow X \cup y, y_{start} = x_{start} \wedge y_{self} = r \wedge y_{goal} = x_{goal} \wedge y_{dist} = x_{dist} + \delta(a)$ 
28:       end if
29:     end for
30:   end for
31:    $X \leftarrow X \setminus \{y \mid y \in X \wedge y_{self} = x_{self}\}$ 
32: end while
33: return  $P$ 

```

発データ^{*2}から、論理式変換器 Boxer^{*3}を用いて作成されたもので、777個の観測からなる。それぞれの観測は平均して27.9個のリテラルによって構成されている。

背景知識については、WordNet [3]から抽出した289,655個の公理と、FrameNet [13]から抽出した7,558個の公理を用いて論理式の集合を作成した。この背景知識における述語の異なり数は269,725であった。

5.2 実験設定

本実験では、井之上らの手法と提案手法のそれぞれについて、上述のデータセットに対する求解時間を比較する。以降、前者を **Baseline**、後者を **A*-based** と表記することにする。パラメータに関して、Baseline については $depth_{max} = 3, 4$ の2通りで実験を行い、A*-based については $dist_{max} = 4, 5, 6$ の3通りについて実験を行った。また、Baseline との比較を行いやすくするために、A*-based における経路コスト関数 $\delta(a)$ の定義は $\delta(a) = 1.0$ とした。また、A*-based における経路コストの推定値については、事前に全ての述語ペアについて計算したものを使用した。これらの計算に要した時間についても次節で報告する。

実験には、CPU 2.2GHz (コア数 48)、RAM 256GB、OS Ubuntu12.10 (64bit) の計算機を用いた。ILP ソルバには Gurobi Optimizer^{*4} を使用し、Gurobi の計算に用いるスレッド数は8とした。また、ILP 問題の最適化において120秒以上かかるものについてはタイムアウトとし、結果から除外した。

5.3 結果

実験の結果を表1に表す。表中の各行について、**Objective** は解仮説のコストの平均値、**Rate of objective** は解仮説のコストにおける Baseline ($depth_{max} = 4$) との相対値の平均値を示す。

また、**# of literals** は潜在仮説集合に含まれるリテラルの数の平均値、**# of chains** は潜在仮説集合に含まれる逆向き推論の数の平均値、**# of unifications** は潜在仮説集合に含まれるリテラル間の単一化の数の平均値を示す。

Time (P-Gen) は潜在仮説集合の生成に要した時間の平均値、**Time (ILP-Conv)** は潜在仮説集合を整数線形計画問題に変換するのに要した時間の平均値、**Tim (ILP-Solve)** は整数線形計画問題の最適解を導出するのに要した時間の平均値、**Time (ALL)** は推論全体でかかった時間の平均値、**Time (d^*)** は A*-based における全述語ペアに対する推定値の計算にかかった時間を示す。

of timeout は、タイムアウトした問題の個数を表す。また、表中の平均値は、全ての実験設定においてタイムア

^{*2} <http://pascallin.ecs.soton.ac.uk/Challenges/RTE2/>

^{*3} <http://svn.ask.it.usyd.edu.au/trac/candc>

^{*4} <http://www.gurobi.com/>

	Baseline ($depth_{max} = 4$)	Baseline ($depth_{max} = 3$)	A*-based ($dist_{max} = 6$)	A*-based ($dist_{max} = 5$)	A*-based ($dist_{max} = 4$)
Objective	278.5	279.8	279.8	280.6	282.6
Rate of objective	1.0	1.0050	1.0051	1.0082	1.0152
# of literals	1579	1059	233	176	65
# of chains	1528	1013	189	136	27
# of unifications	758	395	114	69	39
Time (P-Gen)	0.6	0.2	0.8	0.5	0.2
Time (ILP-Conv)	1.6	0.2	0.04	0.02	0.004
Time (ILP-Solve)	47.4	15.6	3.0	1.3	0.4
Time (ALL)	49.5	15.9	3.8	1.7	0.6
Time (d^*)	-	-	1514	407	106
# of timeout	125	48	16	12	7

表1 先行研究との比較実験の結果。時間を表す項の値はいずれも単位は秒である。

ウトしなかった問題だけを計算の対象とした場合の値である。そのため、Rate of objective は各設定の探索範囲の狭さを表す指標として考えることができる。

以下にこの結果に対する考察を述べる。まず、表1の# of literals, # of chains, # of unifications の比較より、提案手法によって生成される潜在仮説集合の規模が、Baseline によって生成される潜在仮説集合の規模よりも、遥かに小規模であることが確認できた。さらに、Time (ILP-Solve) の比較より、ILP 問題の最適化にかかる時間も少なくなっており、ほぼ同じ探索範囲の広さを持つ Baseline ($depth_{max} = 3$) と A*-based ($dist_{max} = 6$) の間では、5 倍程度の高速化 (15.6 秒から 3.0 秒) が実現できた^{*5} ことが確かめられた。

また、表1の# of timeout より、タイムアウトした問題の数が3分の1程度に減っている (48 個から 16 個) ことも確かめられた。

A*-based における経路コスト推定値の計算に要する時間は、 $dist_{max}$ の値に応じて指数的に増大していることが確かめられたが、実用的には、 $dist_{max}$ の値は 3~6 程度で十分であると考えることや、前処理として計算可能であることから、推定値の計算コストが本手法のボトルネックにはならないと考えられる。

6. 結論

仮説推論は、文章に明示されていない情報の顕在化を行うための有望な枠組みと考えられてきた一方、背景知識や観測の規模に対して指数的に増大する計算時間が、実問題への応用を阻んできた。本稿では、仮説推論の計算速度の問題を改善するために、最新の先行研究である井之上ら [17] の手法を改善する手法を提案した。より具体的には、潜在仮説集合の生成に A*探索を適用することにより、元の手法に比べて、潜在仮説集合の規模が大幅に削減され、それに伴って ILP 問題の求解速度が大幅に改善されること

^{*5} なお、表1の# of Time (P-Gen) より、A*-based における潜在仮説集合の生成時間が Baseline よりも長いのが、その原因は、推定値をデータベースから参照する際にかかるディスク I/O の速度だと考えられる。

を示した。

今後は、距離の推定値を、異なる高速化のアプローチに応用することも検討している。より具体的には、推定値を用いて観測リテラル間の単一化の可能性を判別し、観測を互いに独立に推論可能な単位に分割することで、推論を並列化することを予定している。

謝辞

本研究は、JST 戦略的創造研究推進事業「さきがけ」および JST 戦略的創造研究推進事業「CREST」から部分的な支援を受けて行われた。

参考文献

- [1] Blythe, J., Hobbs, J. R., Domingos, P., Kate, R. J. and Mooney, R. J.: Implementing Weighted Abduction in Markov Logic, *Proceedings of IWCS*, Oxford, UK, pp. 55–64 (2011).
- [2] Chambers, N. and Jurafsky, D.: Unsupervised Learning of Narrative Schemas and their Participants, *Proceedings of ACL*, pp. 602–610 (2009).
- [3] Fellbaum, C.(ed.): *WordNet: an electronic lexical database*, MIT Press (1998).
- [4] Hobbs, J. R., Stickel, M., Martin, P. and Edwards, D.: Interpretation as Abduction, *Artificial Intelligence*, Vol. 63, pp. 69–142 (1993).
- [5] Inoue, N. and Inui, K.: Large-scale Cost-based Abduction in Full-fledged First-order Predicate Logic with Cutting Plane Inference, *JELIA*.
- [6] Inoue, N. and Inui, K.: ILP-Based Reasoning for Weighted Abduction, *Proceedings of AAAI Workshop on Plan, Activity and Intent Recognition* (2011).
- [7] Inoue, N. and Inui, K.: An ILP Formulation of Abductive Inference for Discourse Interpretation, Technical Report 3 (2011-09-09).
- [8] Inoue, N., Ovchinnikova, E., Inui, K. and Hobbs, J.: Coreference Resolution with ILP-based Weighted Abduction, *COLING*, pp. 1291–1308 (2012).
- [9] Ovchinnikova, E., Montazeri, N., Alexandrov, T., Hobbs, J. R., McCord, M. and Mulkar-Mehta, R.: Abductive Reasoning with a Large Knowledge Base for Discourse Processing, *Proceedings of IWCS*, Oxford, UK, pp. 225–234 (2011).
- [10] Penas, A. and Hovy, E.: Filling knowledge gaps in text for machine reading, *Proceedings of COLING: Posters*, pp. 979–987 (2010).

- [11] Raghavan, S. and Mooney, R. J.: Bayesian Abductive Logic Programs, *Proceedings of STARAI*, pp. 82–87 (2010).
- [12] Richardson, M. and Domingos, P.: Markov logic networks, *Machine Learning*, pp. 107–136 (2006).
- [13] Ruppenhofer, J., Ellsworth, M., Petruck, M., Johnson, C. and Scheffczyk, J.: FrameNet II: Extended Theory and Practice, Technical report, Berkeley, USA (2010).
- [14] Schoenmackers, S., Davis, J., Etzioni, O. and Weld, D.: Learning First-order Horn Clauses from Web Text, *Proceedings of EMNLP*, pp. 1088–1098 (2010).
- [15] Singla, P. and Domingos, P.: Abductive Markov Logic for Plan Recognition, *Proceedings of AAAI-11*, pp. 1069–1075 (2011).
- [16] 杉浦 純, 井之上直也, 乾健太郎: 説明生成に基づく談話構造解析の課題分析, 言語処理学会第 18 回年次大会論文集, pp. 115–118 (2012).
- [17] 井之上直也, 乾健太郎, Ovchinnikova, E., Hobbs., J. R.: 大規模世界知識を用いた仮説推論による談話解析の課題と対策, 言語処理学会第 18 回年次大会論文集, pp. 119–122 (2012).