

言語処理のための仮説推論エンジン Phillip

山本 風人 井之上 直也 乾 健太郎

東北大学 情報科学研究科

{kazeto, naoya-i, inui}@ecei.tohoku.ac.jp

1 はじめに

仮説推論は、与えられた観測集合に対する最良の説明を見つける推論である。この種の推論は、談話解析やプラン認識を含む様々な人工知能分野のタスクにおいて用いられてきた [1, 7, 13, 11]。

これらの研究の中で代表的な枠組みが Hobbs ら [3] 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 (Abduction) 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仮説推論を用いて解釈を行った結果の例を、図 1 に示す。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がわかる。

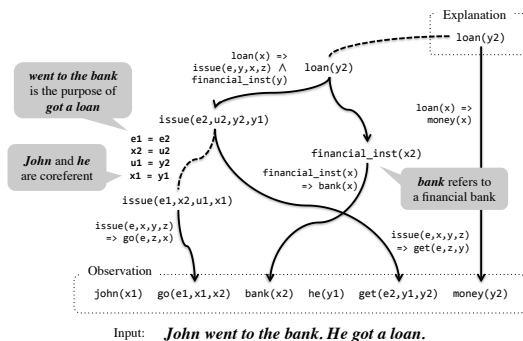


図 1: 仮説推論による言語理解の例

仮説推論における課題のひとつに、計算コストが挙げられる。一階述語論理における仮説推論、あるいはそれと同等の表現力を持つ論理推論では、解を得るのに要する計算量が非常に大きく、実問題への適用における問題の一つとなっている。これに対し井之上ら [4, 13] は一階述語論理における仮説推論を整数線形計画問題 (ILP) に変換する手法を提案し、その手法によって仮説推論の計算コストが著しく改善できることを示した。また、井之上らの研究をうけて我々 [10] は、A*探索に基づく手法によって解候補を効率的に列挙する手法を提案し、井之上らの手法と我々の手法を組み合わせることによって、さらに仮説推論の計算が効率化できることを示した。しかしながら、計算コストの問題は未だ完全に解決されたとはいえないのが実情である。仮説推論の解の探索空間は背景知識の規模に応じて指数関数的に増大す

るため、大規模知識による仮説推論においては、上述のシステムであっても解が得られない状況が存在しうる。

このような背景を踏まえ本稿では、仮説推論を言語処理タスクに適用する場合を前提として、文を論理表現で表す際にそれらの意味表現に内在する性質に着目し、それらを明示的な制約として定義するための機能を提供するような推論システム Phillip を構築する。また、構築したシステムが、従来手法よりも遥かに高速な推論を実現できていることを実験的に示す。なお、本稿で示すシステムは既に一般に利用可能なソフトウェアとして公開されている¹。

本稿では、まず仮説推論と、先行研究について述べたあと (2 節)、提案手法について説明する (3, 4 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 (5 節)、最後に今後の展望を述べる (6 節)。

2 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 はホーン節の集合、 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。

一般には、上の条件を満たすような H は複数存在する。本稿ではそのような H を候補仮説と呼び、候補仮説に含まれる各リテラルを要素仮説と呼ぶこととする。また、可能な全ての候補仮説に対する要素仮説の集合を潜在仮説集合と呼ぶ。ここで、候補仮説は潜在仮説集合の部分集合であるため、潜在仮説集合は探索空間を定義するものとして見做することができる。

仮説推論の目的は、何らかの評価指標のもとで最良の候補仮説 \hat{H} を見つけることである。この \hat{H} を解仮説と呼び、次式のように表せる。

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

ここで、 \mathbb{H} は可能な候補仮説の集合、 $Eval$ は候補仮説 H の蓋然性を表す何らかの評価値を返すような関数を表し、このような関数を評価関数と呼ぶ。評価関数としては、様々な関数が提案されている。[3, 9, 13, 8]

上述のように、候補仮説はそれぞれ潜在仮説集合の部分集合として見做することができる。潜在仮説集合は

¹<http://github.com/kazeto/phillip>

以下の二つの操作を観測に対して逐次適用することによって生成される：(i) 後ろ向き推論は、背景知識 $P_1(x) \wedge P_2(x) \wedge \dots \wedge P_n(x) \Rightarrow Q(x) \in B$ 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、潜在仮説集合に追加する処理である。例えば、 $O = \exists x A(q(x))$, $B = \forall x (p(x) \Rightarrow q(x))$ が与えられた場合は、リテラル $p(A)$ が潜在仮説集合に追加される。(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。例えば、 $O = \exists x, y (p(x) \wedge p(y) \wedge q(y))$ が与えられた場合、 $x = y$ が潜在仮説集合に追加される。

3 言語的知見に基づく制約の定義

文や言語表現を論理式で表す上で、その論理表現には、言語的な知見に起因した様々な非明示的な性質が内在する。従来のシステムではそれらの性質を明示的な制約として定義するための枠組みが提供されていなかったために、論理的には可能だが言語的には正しくないような仮説が数多く生成されてしまう事態に陥っていた。例えば図2のような、各述語に対する主格の

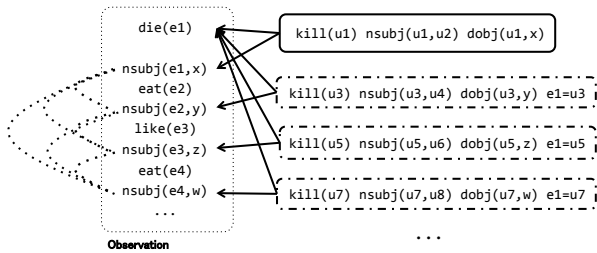


図2: 言語的に正しくない仮説群の例

情報をリテラル *nsubj* で表すような論理表現を考えると、従来のシステムでは、図のようにして、あらゆる *die* と *nsubj* の組み合わせに対して後ろ向き推論を適用し、あらゆる *nsubj* のペアに対して単一化を適用してしまう。しかし、これらのうち言語的に正しいと考えられるのは、*die(e1)*, *nsubj(e1, x)* に対する後ろ向き推論と、*nsubj(e2, y)*, *nsubj(e4, w)* に対する単一化のみであり、それら以外の仮説については、例えば *nsubj(e1, x)*, *nsubj(e2, y)* の単一化は「あるイベント $e1 (= e2)$ が “die” かつ “eat” で表される」という仮説を導いているなど、言語的には意味の無い推論になってしまっている。このような推論は、解の言語的な正しさという観点でも、計算コストの軽減という観点でも、解候補からは除外できることが望ましい。

本稿では、このような問題に対して、後ろ向き推論と単一化のそれぞれに対して、変数の共有関係に関する制約を与えることを考える。具体的には以下の様にして制約を記述する：

1. $(\Rightarrow (\wedge (\text{kill } e1) (\text{nsubj } e1 \ x) (\text{dobj } e1 \ y)))$
 $(\wedge (\text{die } *e2) (\text{nsubj } *e2 \ y)))$
2. $(\text{unipp } (\text{nsubj } * \ .))$

前者の例は、後ろ向き推論に対する制約の記述例である。この公理を後ろ向き推論に用いる際に、*から始ま

る変数名を持つ変数については、対応した引数が適用先のリテラルの引数が一致していない場合（例、 $\text{die}(e1) \wedge \text{nsubj}(e2, x)$ ）には後ろ向き推論の適用を制限する。これにより、述語と格を帰結部に持つような公理の使用において、異なる動詞に対する格を推論に用いるような推論を制限することが可能となる。例えばこの制約を図2の例に適用した場合、 $\text{kill}(u1)$ を導く推論のみが仮説される。

後者の例は、単一化に対する制約の記述例である。この例では、主格を表す述語 *nsubj* について、“一つ目の引数の等価性が既に単一化されている状況でなければ単一化を制限する” という制約を定義している。これにより、例えば異なる動詞に対する *nsubj* の単一化を制限しながら、同義語などの動詞そのものが同一のイベントを表しうる状況においては *nsubj* の単一化も可能にする、柔軟な制約を定義することが可能となる。例えばこの制約を図2の例に適用した場合は、 $e1 = e4$ を導く単一化のみが仮説される。

別の問題として、A*探索における関連度の推定精度に関する問題がある。我々の先行研究 [10] において、背景知識における述語間の関連度を予め推定しておくことで、A*探索に基づいた探索を行い、効率的に潜在仮説集合を生成する手法を提案した。しかし、格関係や前置詞などを表すような機能的なリテラルを含む推論においては、関連度の計算時にそれらのリテラルを経由することで殆どのリテラル対が互いに繋がることから、関連度の推定精度が著しく低下してしまうという問題が生じる。A*探索における推論効率には、関連度の精度に依存するため、精度の低下はそのまま計算コストの増加に繋がる。

これに対し、本システムでは、指定した述語を関連度の計算から除外（ストップワード）するような機能を提供する。ストップワードとして格関係や前置詞などを表すような機能的なリテラルを指定することにより、解の最適性を保持したまま、関連度の精度の低下を防ぐことが可能になる。なお、より具体的には、背景知識に含まれる各公理について、公理の一辺がストップワードのみで構成されなければよい。

4 実現不可能な推論の除外

先行研究 [10] において、潜在仮説集合の規模を削減することで、推論全体の計算量が軽減できることが確かめられた。本節ではこの事を踏まえ、解に含まれ得ない推論を更に積極的に探索し、潜在仮説集合をより小さくすることを目指す。より具体的には、潜在仮説集合の中には仮説としては存在するものの実際には実現不可能な推論が含まれることがある。このような推論が仮説に含まれることは無いため、それらを予め潜在仮説集合から除外することによって、解の最適性を保持したまま推論全体の計算量を軽減することが出来る。

実現不可能な推論の例を図3に示す。 $\text{police}(x)$ と $\text{criminal}(x)$ の間に排他関係が定義されている場合を考えた時、この推論によって仮説されるリテラル $\text{police}(\text{Tom})$

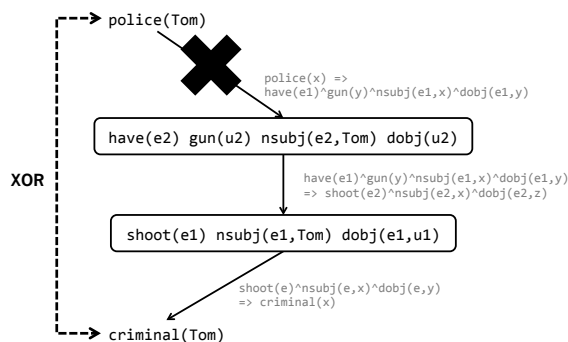


図 3: 実現不可能な推論の例

とその祖先 $criminal(Tom)$ の間には無条件で排他関係が成立することから、この推論は実現不可能であり、除外できると見做せる。

実際の実装では、各リテラルの祖先をそれぞれ保持しておくことで、実現可能性の判定処理を高速化している。後ろ向き推論や単一化を行う際に、操作の対象となるリテラル群の祖先の和集合をとり、それが矛盾を含むかどうかを判定することで、効率的に判定が行える。

5 評価実験

5.1 データセット

観測としては、Winograd Schema Challenge[5]に含まれる文章に対して Stanford Core NLP [6] を用いて構文解析を行い、その結果を論理式に変換したデータを用いた。今回は Winograd Schema Challenge のデータから無作為に 25 文を選び、25 個の観測へ変換した。それぞれの観測は平均して 49 個のリテラルによって構成される。

背景知識には、WordNet[2] から抽出した公理と、杉浦ら [12] が ClueWeb12² から構築した、約 2.3 億の共参照事例からなるデータセットを用いて構築した公理を用いた。背景知識に含まれる公理の総数は 1,111,219 個であり、そのうち WordNet から抽出した公理の数は 290,487、共参照事例から生成した公理の数は 820,732 個であった。

5.2 設定

実験設定としては、図 4 下部に示す 10 個の設定で推論を行い、各観測に対する求解時間を比較した。ここで KB size は背景知識の規模に対する倍率を示す。例えば KB/2 の設定では、背景知識に含まれる公理のうち半分をランダムに除外したものを背景知識として用いた。CH, UNI, SW, CA の各行はそれぞれ機能の使用の是非を表し、CH は後ろ向き推論に対する制約、UNI は単一化に対する制約、SW は関連度計算におけるストップワードの指定、CA は実現不可能な推論の除外に対応している。推論の深さは全て 2 段で行ったが、Dep3 に限り深さ 3 で推論を行った。

実験には、CPU Opteron2435 2.6GHz (コア数 12)、RAM 64GB、OS Ubuntu14.04 (64bit) の計算機を用いた、ILP ソルバには Gurobi Optimizer³ のバージョン 5.6 を使

用し、Gurobi の計算スレッド数は 8 とした、また、推論に 30 分以上かかるものについてはタイムアウトとして扱った。

5.3 実験結果

実験の結果を図 4 に表す⁴。図中の各点は各観測の最適解の導出に要した時間を表し、緑色の横線はタイムアウトの時間 (1800 秒) に対応している。以下にこの結果に対する考察を述べる。

まず、既存手法 [10] との比較を行う。Baseline が既存手法、Phil が提案手法に対応する。Baseline が 25 問全てタイムアウトしている一方で、Phil はそれらの問題に対する解をいずれも 1 分以内で得ることができていることから、大規模知識を用いた仮説推論においては、我々のシステムが従来システムよりも遥かに高速に解が得られていることが分かった。

また、背景知識の規模に応じた推論速度の変化については、Phil, KB/2, KB/4, KB/8 を比較することで確認できる。仮説推論の計算コストは 1 節で述べたように背景知識の規模に応じて指数関数的に増大するが、本実験でもその関係は成り立っていることが分かる。

次に、本稿で提案した各種機能ごとの推論速度の変化について検証する。Phil, NonCH, NonUNI, NonSW, NonCA の比較から、今回の実験において計算の効率化に最も貢献しているのは、関連度計算においてストップワードを指定することであることが確かめられた。

最後に、Dep3 の結果について検証する。まず、Dep3 における解析結果について詳しく検証すると、解析時間の大半は潜在仮説集合の生成に費やされていたことと、解析時間の爆発的増大に関わらず、潜在仮説集合の規模は Phil における解析結果の数倍程度に留まっていることが確認できた。このことから、本システムにおけるボトルネックは、後ろ向き推論を行う際の、潜在仮説集合に対して適用可能な公理を列挙する処理であることがわかる。

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。

このような問題に対し、本稿では、言語処理タスクにおける仮説推論の適用に特化した仮説推論エンジンを構築し、その概要を述べた。これは、先行研究 [10] を土台として、さらに大規模知識を用いた仮説推論における計算コストの爆発的な増加を抑えるための種々の機能を追加している。また、実験を通じて、構築したシステムが従来システムよりも遥かに高速であることを示した。

今後の展望としては、潜在仮説集合の生成処理のさらなる高速化を進めていく。本稿の実験により確かめられ

²<http://lemurproject.org/clueweb12>

³<http://www.gurobi.com/>

⁴NonCH について、執筆時点の実装では解の探索空間が不正に小さくなってしまったため、本稿の結果は参考記録とする。ただし、正しい実装においても推論時間自体はほぼ変わらないと考えられる。

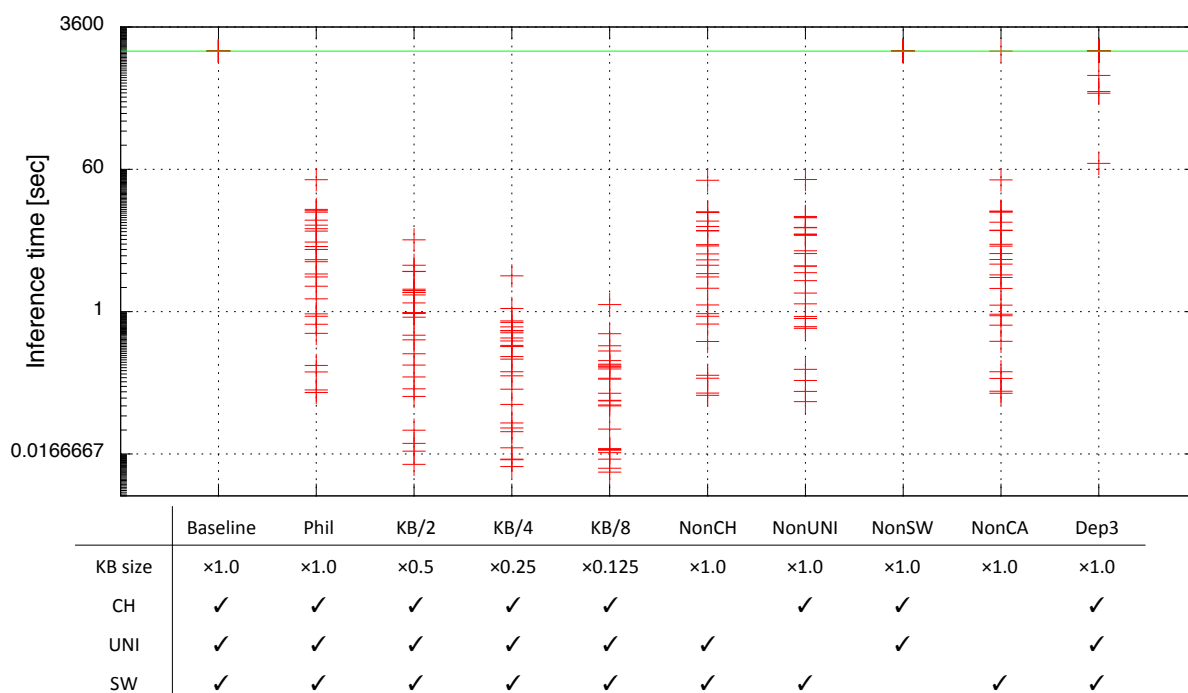


図 4: 実験結果

たように、本システムにおけるボトルネックは潜在仮説集合を生成する際の適用可能な公理を列挙する処理であった。つまり、この処理を効率化することによって、仮説推論を更に高速化できると考えられる。解決策としては、現状では以下の2通りの案を考えている。ひとつは、公理を背景知識から検索してくる処理を高度化することである。現状では単一の述語でインデックス化しているために、検索によって得られる公理の数が肥大化しがちであり、結果として適用可能性の検証に多大な計算コストを払うことになってしまっている。これをより細かい検索を可能にすることで、計算量の低減を図ることが考えられる。もうひとつは、探索空間の爆発が起こらないように背景知識の構成をコントロールすることである。一般に仮説推論の探索空間は背景知識の構造に強く依存するため、探索空間の爆発が生じないように背景知識の構成をコントロールすることが肝要となる。例えば今回用いた背景知識に対しては、A*探索での各公理の距離値を公理のリソースごとに調整することや、ストップワードをさらに追加する、“have”や“make”などといった機能動詞に対する公理抽出の方法を検討するなどの案が考えられる。どのような構成によって探索空間をコントロールするかはタスクや背景知識に依存しているために一概には言えないものの、そのコントロールを実現するための手段を提供することは、今後システム側が果たすべき役割であると考えられる。

謝辞

本研究は、JST 戦略的創造研究推進事業「CREST」および文部科学省科研費（23240018）から部分的な支援を受けて行われた。

参考文献

- [1] J. Blythe, J. R. Hobbs, P. Domingos, R. J. Kate, and R. J. Mooney. Implementing Weighted Abduction in Markov Logic. In *Proceedings of IWCS*, pp. 55–64, Oxford, UK, 2011.
- [2] C. Fellbaum, editor. *WordNet: an electronic lexical database*. MIT Press, 1998.
- [3] J. R. Hobbs, M. Stickel, P. Martin, and D. Edwards. Interpretation as abduction. *Artificial Intelligence*, Vol. 63, pp. 69–142, 1993.
- [4] N. Inoue and K. Inui. ILP-Based Reasoning for Weighted Abduction. In *Proceedings of AAAI Workshop on Plan, Activity and Intent Recognition*, 2011.
- [5] Hector J. Levesque. The winograd schema challenge. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*. AAAI, 2011.
- [6] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55–60, 2014.
- [7] 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 *Proceedings of IWCS*, pp. 225–234, Oxford, UK, 2011.
- [8] S. Raghavan and R. J. Mooney. Bayesian Abductive Logic Programs. In *Proceedings of STARAI*, pp. 82–87, 2010.
- [9] P. Singla and P. Domingos. Abductive Markov Logic for Plan Recognition. In *Proceedings of AAAI-11*, pp. 1069–1075, 2011.
- [10] K. Yamamoto, N. Inoue, K. Inui, Y. Arase, and J. Tsujii. Boosting the efficiency of first-order abductive reasoning using pre-estimated relatedness between predicates. *International Journal of Machine Learning and Computing*, Vol. 5, No. 2, pp. 114–120, 2015.
- [11] 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.
- [12] 杉浦純, 井之上直也, 乾健太郎. 共参照解析における事象間関係知識の適用. 言語処理学会第 20 回年次大会論文集, pp. 713–716, 2014.
- [13] 井之上直也, 乾健太郎, E. Ovchinnikova, Jerry R. Hobbs. 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.