

知識ベース補完を用いた高階論理推論のための自動公理生成

吉川 将司¹ 峯島 宏次² 能地 宏¹ 戸次 大介²
 奈良先端科学技術大学院大学¹ お茶の水女子大学²

yoshikawa.masashi.yh8@is.naist.jp, mineshima.koji@ocha.ac.jp,
 noji@is.naist.jp, bekki@is.ocha.ac.jp

1 はじめに

含意関係認識 (Recognizing Textual Entailment; 以下 RTE) などの深い意味処理を扱うタスクでは、大量の知識データが必要である。高階論理による RTE システムである `ccg2lambda`[6] は、WordNet[2] や VerbOcean[1] などの知識ベースから証明に必要な知識を検索し、論理式の形で公理として利用することで大きな性能改善を示した [8] (§2.1 参照)。しかし、これらのような大規模な知識ベースを離散的なデータ構造として保持しておくことは、使用するメモリ量や計算効率の面からは問題であり、性能の改善のために知識データを拡充しようとする、問題を処理する効率性を失ってしまうというジレンマがある。

一方で、近年機械学習を用いた知識ベース補完 (Knowledge Base Completion) の技術の発展が目覚ましい [11, 10]。これらの研究の多くでは、計算効率の良さが重視されており、知識ベースの entity s, o と relation r はパラメータ n の次元のベクトルとして表現され、三つ組 (s, r, o) が事実として真であるか偽であるかの判定は $O(n)$ のスコア計算で行われる。これらの手法は、WordNet を基に作られたベンチマークにおいてすでに高い精度を示しており、上述の公理生成における離散的な知識ベースの検索を置き換えることが期待できる。さらに、これらの研究は、既存の知識ベースに欠けた事実 (三つ組) をスコア関数で補完することに動機づけられているが、このような潜在的な知識は当然 RTE においても有用である。

そこで本研究では、`ccg2lambda` を知識ベース補完による自動公理生成で拡張する。具体的には、これまでの WordNet を検索することで行っていた公理生成を、上述のスコア関数で置き換えることにより、RTE の問題をより高精度かつ高速に処理可能にすることを目指す。そのため、論理推論に有用なスコア関数を学習するために適した新たなデータセットを WordNet から

構築する。本研究ではまた、そうして構築したデータに VerbOcean からの知識を加えることで、メモリ量、計算時間における効率を損なわずに RTE における性能向上を導くデータ拡張が可能であることを示す。

加えて、論理推論をさらに効率的に行うための改良を行う。既存研究 [8] では、新たな公理を追加するために定理証明器 (Coq[5]) を一度終了し、公理を追加した後に定理証明を再実行する必要があった。本研究では Coq を実行しながら同時に WordNet の検索や知識ベース補完モデルによる評価を実行できる Coq の abduction tactic を開発する。

SICK[7] データセットで実験を行い、ベースラインに対して精度においては僅かに下回る結果となったが、処理速度において大きな改善を示した。

2 既存研究

2.1 `ccg2lambda`

`ccg2lambda`[6] は、RTE の前提文 (P) と帰結文 (H) を、統語構造と意味表示が透明な関係を持つ組み合わせ範疇文法 (Combinatory Categorical Grammar; CCG)[3, 13] の構文解析器によって解析した後、 λ 計算による意味合成によって高階述語論理式に変換する (図 1a, b)。それらの論理式に対し、含意 (もしくは矛盾) が成り立つかを調べるため、含意の論理演算子で結合し ($P \rightarrow H, P \rightarrow \neg H$)、証明支援システムである Coq[5] で自動推論を行う (図 1c)。

Coq は tactic と呼ばれる命令を組み合わせることで人による定理証明を支援するシステムであるが、firstorder tactic では一階述語論理の論理式に対する自動の定理証明を行うことができる。またその際に高階の自然言語の論理表現についてはその推論規則を公理として登録しておくことで高階の推論にも対応できる。

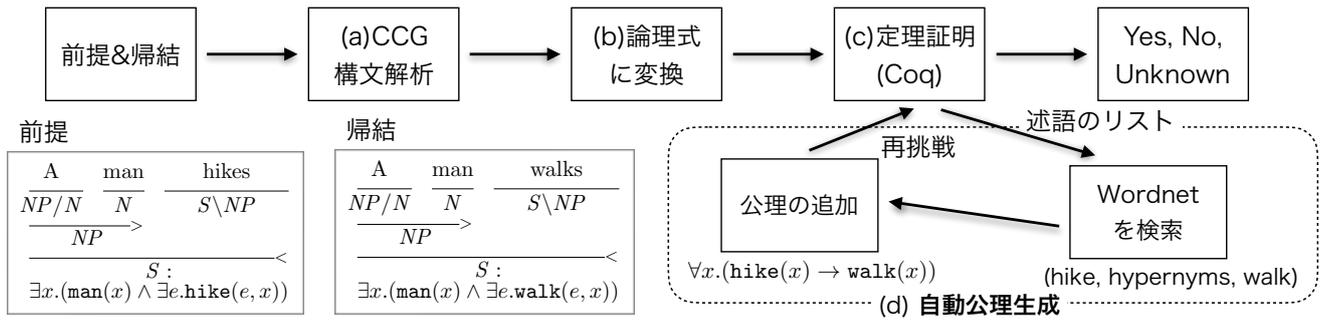


図 1: ccg2lambda の処理過程. 前提 (P) と帰結 (H) を CCG パーザで解析し (a), 論理式へ変換する (b). それらを含意演算子で結合し ($P \rightarrow H, P \rightarrow \neg H$), Coq で推論を行う (c). 証明に失敗した場合, 公理生成を試みる (d).

既存研究 [8] による拡張では, まず Coq で RTE の問題の証明を行い, 失敗した場合に証明を完了するために有用な公理の探索を行う (abduction). WordNet[2] と動詞間の関係を記述した VerbOcean[1] から語彙間知識を検索し, 論理式に変換してから公理として宣言を行い, 再度証明を試みる (図 1d). ここでの問題点として, WordNet 等を検索するために Coq プログラムを一度終了する必要がある, 全体の処理速度が下がってしまうことが挙げられる. 本研究では Coq を実行しながら同時に WordNet の検索や知識ベース補完を実行できる abduction tactic の開発を行う (§3.2).

2.2 知識ベース補完

本研究では知識ベース補完のモデルとして ComplEx[10] を使用する. ComplEx では三つ組 (s, r, o) に対して, 複素数¹を用いてその関係をモデル化する:

$$f(s, r, o) = \sigma(\text{Re}(\langle e_s, e_r, \bar{e}_o \rangle)). \quad (1)$$

ここで, $e_s, e_r, e_o \in \mathbb{C}^n$, $\langle x, y, z \rangle = \sum_i x_i y_i z_i$ であり, n は埋め込みベクトルの次元数である.

3 提案法

提案法では, ccg2lambda による論理推論において前提と帰結の間に語彙的な隔たりが存在する場合に, 知識ベース補完のスコア関数を用いて新たな公理を追加する. 具体的に, ComplEx モデル (式 1) で前提と帰結に存在する論理述語のペアをすべての関係 $r \in R = \{\text{synonym, hypernym, antonym, hyponym}\}$ について評

¹ここで複素数の集合を \mathbb{C} で示し, $x \in \mathbb{C}$ について, その実部を $\text{Re}(x)$, 複素共役を \bar{x} と表す.

関係 r	生成する公理
synonym, hypernym	$\forall \vec{x}.(s(\vec{x}) \rightarrow o(\vec{x}))$
例. (make, r, build) $\leadsto \forall \vec{x}.(\text{make}(\vec{x}) \rightarrow \text{build}(\vec{x}))$	
antonym	$\forall \vec{x}.(s(\vec{x}) \rightarrow \neg o(\vec{x}))$
例. (parent, r, child) $\leadsto \forall \vec{x}.(\text{parent}(\vec{x}) \rightarrow \neg \text{child}(\vec{x}))$	
hyponym	$\forall \vec{x}.(o(\vec{x}) \rightarrow s(\vec{x}))$
例. (talk, r, advise) $\leadsto \forall \vec{x}.(\text{advise}(\vec{x}) \rightarrow \text{talk}(\vec{x}))$	

表 1: 三つ組 (s, r, o) と r に関して生成される公理. 述語の引数の数・型は ccg2lambda の意味解析時に決定し, 公理生成時はそれを用いる.

価し, スコアが閾値 θ 以上のものを表 1 に従い論理式に変換し, 公理として宣言する.

以下, §3.1 で公理生成を主眼においたデータセット構築手順を, §3.2 で Coq 実行時に動的に公理生成を行う abduction tactic について述べる.

3.1 データセットの作成

自動公理生成のためのデータセットを WordNet から構築する上での問題点として, WordNet では (hypernym, hyponym など) 多くの関係が見出し語 (lemma) の集合である同義語集合 (synset)²の間で定義されていることがある. ccg2lambda では文から論理式に変換する際に表層形から見出し語に正規化しているため, 見出し語間の知識データが必要である.³ 以下, この隔たりを考慮しながら各 $r \in R$ について WordNet を走査して三つ組を収集し, データセットを構築する.

まず synonym について, 同義語集合 s_1 に対して WordNet 上で also_sees, verb_groups, similar_tos の関係にある同義語集合 s_2 を s_1 と同

²以下, 簡略のため同義語集合 s に見出し語 l が含まれることを集合論の記法を用い $l \in s$ と示す.

³このことは, 後に見出し語間で関係を定義している VerbOcean や, 生テキストなど他のデータを利用した拡張を行う上でも重要である.

義語 (synonym) の関係にあるとみなし, すべての $l_1 \in s_1, l_2 \in s_2$ の組みについて $(l_1, \text{synonym}, l_2)$ としてデータセットに含める. また, ある $l \in s_1$ について, $l \in s_2$ であるような任意の s_2 も同義語とみなし同様の処理を行う. 上位語 (hypernym) の関係にある s_1 と s_2 についても, synonym と同様にすべての見出し語の組を hypernym としてデータに含めた. hypernym は, 推移律が成り立つと考え⁴, その推移閉包から得られる関係についてもデータセットに含める. hyponym についても hypernym と同様の処理を行う. 反意語 (antonym) については WordNet において見出し語間で定義されているため, それらをデータセットに含めた.

このようにして取り出した三つ組の量は多く, また現在の RTE タスクを解く上で有用ではない見出し語も多く含まれるため, 見出し語リスト⁵ を用いて, 三つ組 (l_1, r, l_2) について l_1, l_2 の両方がリストに含まれるもの以外は取り除いた. 結果として構築されたデータセットは 2,588,186 の三つ組からなり, entity の総数は 40,488 となった. 開発用に 5,000 組を取り, それ以外すべて学習に用いる.

VerbOcean は, 見出し語の間に関係を定義しているため, 前述の見出し語リストによるフィルタを行った後, VerbOcean 上で定義された関係から R へのマッピングを行った.⁶ VerbOcean からは 24,953 組の三つ組を抽出した.

3.2 abduction tactic による公理生成

既存研究 [8] では, 公理生成を行うために Coq のプログラムを一度終了する必要があった. 本研究では, より効率よく論理推論を行うために, Coq を実行しながら同時に WordNet の検索や知識ベース補完のモデルを実行できる abduction tactic を開発した. tactic の具体的な動作を図 2 に示す. abduction が実行されたとき, 動作途中のコンテキスト (仮定の集合) とゴール (証明が必要な命題) の間のすべての論理述語の組みを取り出し, 別プロセスで動作する Python サーバに送信する. Python サーバはそれぞれの組について, すべての $r \in R$ に対し ComplEx (式 1) で評価し, 閾値 θ 以上の三つ組を Coq に送り返す. Coq 側で表 1

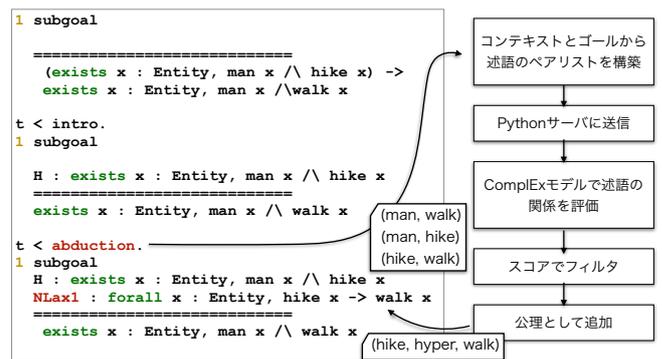


図 2: abduction tactic の実行例.

手法	正答率	適合率	再現率	処理速度
ベースライン (WN)	83.55	97.20	63.63	10.92
+ VO	83.84	96.70	64.66	11.06
abduction tactic	83.60	97.15	63.91	7.41
+ WN	82.82	97.07	62.13	4.63
+ VO	82.95	96.87	62.55	4.63

表 2: SICK データセットにおける実験結果. WN は WordNet, VO は VerbOcean を表す. 処理速度は RTE 問題 1 題あたりに要した時間 (秒) のマクロ平均.

に従い論理式に変換し, 公理として宣言する.

Coq では tactic をつなげて新たな tactic を作ることが可能であるから, 前述の firstorder 等とあわせて abduction も含めて自然言語の高階推論を自動で行う tactic を構成可能である.

スコア関数の評価を Python サーバの別プロセスにすることで, 既存の機械学習ライブラリを利用でき, 機械学習を用いた前提選択 [4] など, より高度な計算を要する定理証明上の問題にも利用可能である. §4 で abduction tactic の有効性を示す.

4 実験

4.1 実験設定

RTE の実験データセットである SICK[7] で提案法の評価を行う. SICK は前提と帰結それぞれ 1 文からなり, そのようなペアが学習/開発/評価用に 4,500/500/4,927 文用意されている. 本研究では学習データについては使用せず, 開発データを処理速度の評価に用いる.

ComplEx のについて, 埋め込みベクトルの大きさ $n = 200$ とし, 学習は以下のロジスティック損失を

⁴(puppy, hypernym, dog),(dog, hypernym, animal) ならば (puppy, hypernym, animal) など.

⁵SICK[7] と SNLI[9] の学習データ, GloVe (<https://nlp.stanford.edu/projects/glove/>) に含まれる見出し語.

⁶具体的に similar, stronger-than → synonym, opposite-of → antonym. それら以外の関係については取り除いた.

Adam で最小化することで行った.

$$\mathcal{L} = \sum_{((s,r,o),t) \in \mathcal{D}} t \log f(s,r,o) + (1-t) \log(1-f(s,r,o)),$$

ここで, \mathcal{D} は学習データであり, §3.1 で作成されたデータの要素なら $t = 1$, それ以外なら $t = 0$ である. 学習を高速化するため, negative sampling の代わりに, 一対多のスコア計算 [11] を行う (詳細は論文参照).

ccg2lambda については公理生成を除いて標準の設定を用いる. 公理生成の閾値 $\theta = 0.4$ とする. ベースラインとして研究 [8] のものを再現し, 評価データにおける正答率/適合度/再現率, 開発データにおける処理速度 (5 回実行したマクロ平均) を報告する.⁷

4.2 実験結果と考察

表2に実験結果を示す. 表において, abduction tactic は §3.2 で開発した tactic を用いるが, 知識ベース補完モデルではなく WordNet を検索して公理生成を行った場合である. +WN で WordNet から作成したデータセットで学習した ComplEx モデルに置き換えたもの, +VO でさらに VerbOcean の三組を学習データに追加した場合を示す. ベースラインと比較して, 証明に要する時間は大きく短縮することができたが, RTE の性能については下がってしまう結果になった. 注目すべき点として+WN と+VO の比較で, 処理速度の変化はない一方で RTE の性能は改善している. VerbOcean によるデータ拡張を行ったが, パラメータ数の増加はしておらず, 計算効率の面では変わらない.

RTE の性能低下に関して, 今回学習したモデルの弱点は単語間の派生関係を捉えにくいことである. 例えば WordNet 上では *sandy* と *sand* は, 派生的な関係にある (derivationally-related) として情報が付与されているが, 今回の学習データにそのような関係は含まなかった. ある語についての形態的な派生をすべて三つ組として表現し, 知識ベース補完の枠組みで扱うことは学習データの爆発的増加につながり困難であると考えられる. 既存研究 [12] では, 知識ベース補完と word2vec 等の単語ベクトルを同時に学習することでより強力なスコア関数の学習が可能なが示されている. 単語ベクトルの学習は, 同義語や派生的な関係にある語の類似度の学習に強力であるから, この問題を解決するために応用することが考えられる.

⁷構文解析と論理式への変換は事前に済ませ, 解析時間には含まない. また, 各 RTE の問題に対して制限時間 100 秒を設定する. これらの実験は 16 コア 2.20 GHz Intel Xeon CPU × 2 を用いて行われた.

5 おわりに

本研究では知識ベース補完による高階論理推論のための自動公理生成の手法を提案した. SICK データセットで実験を行い, 処理速度における大きな向上を示したが, 精度においては課題が残った. 精度面の課題については, 上述の単語埋め込みベクトルとの同時学習による対処を検討する. 今後の課題として, 複数の述語からなる公理の生成 (e.g. $\forall x. \text{have}(x) \wedge \text{fun}(x) \rightarrow \text{enjoy}(x)$) が挙げられる. ccg2lambda の性能の大きな向上にはこの種の公理を正確に生成できることが鍵となっている. そのためには, *have* と *fun* の埋め込みベクトルから構成的な演算で得られたベクトルと *enjoy* のベクトルについて *synonym* 等の関係を学習できるような枠組みを構築する必要がある.

謝辞 この研究は, JST CREST 「ビッグデータ統合利活用のための次世代基盤技術の創出・体系化」領域「知識に基づく構造的言語処理の確立と知識インフラの構築」プロジェクトの支援を受けたものである.

参考文献

- [1] Timothy Chklovski and Patrick Pantel. VerbOcean: Mining the Web for Fine-Grained Semantic Verb Relations. In *Proc. of EMNLP*, 2004.
- [2] George A. Miller. WordNet: A Lexical Database for English. 1995.
- [3] Mark Steedman. *The Syntactic Process*. The MIT Press, 2000.
- [4] Alex A. Alemi *et al.* DeepMath - Deep Sequence Models for Premise Selection. 2016.
- [5] Gilles Dowek *et al.* The COQ Proof Assistant: User's Guide: Version 5.6. INRIA, 1992.
- [6] Koji Mineshima *et al.* Higher-order logical inference with compositional semantics. In *Proc. of EMNLP*, 2015.
- [7] Marco Marelli *et al.* A SICK cure for the evaluation of compositional distributional semantic models. In *Proc. of LREC*, 2014.
- [8] Pascual Martínez-Gómez *et al.* On-demand Injection of Lexical Knowledge for Recognising Textual Entailment. In *Proc. of EACL*, 2017.
- [9] Samuel R. Bowman *et al.* A large annotated corpus for learning natural language inference. In *Proc. of EMNLP*, 2015.
- [10] Théo Trouillon *et al.* Complex embeddings for simple link prediction. In *Proc. of ICML*, 2016.
- [11] Tim Dettmers *et al.* Convolutional 2D Knowledge Graph Embeddings. *AAAI*, 2017.
- [12] Zhen Wang *et al.* Knowledge Graph and Text Jointly Embedding. In *Proc. of EMNLP*, 2014.
- [13] 戸次大介. 「日本語文法の形式理論—活用体系・統語構造・意味合成—」. くろしお出版 日本語研究叢書 24, 2010.