F-028

強化学習による推理小説の犯人推定
Criminal Detection of Mystery Novel by Reinforcement Learning

勝島 修平† 穴田 一†
Shuhei Katsushima Hajime Anada

1. はじめに

近年, 機械学習の発展に伴い, これらの技術への社会的な期待が高まっている一方, 専門家でも結果に対して解釈を与えられない解釈可能性が問題となっている. そんな中、解釈可能性を題材とした推論を行うコンテスト「ナレッジグラフ推論チャレンジ」(以下, チャレンジ 2018)が開催された[1].

既存研究では, 推論過程を評価するためにエキスパートシステムを用いた方法が提案されているが, 途中の自然言語処理により, 元の表現とのずれや, 構造化に多大なコストが掛かり現実的でない. 元の文章表現を変えず行う機械学習による方法が, 大会としても発展性があるとされている.

本研究では, エキスパートシステムの方法ではなく, ナレッジグラフ上のパスの遷移によってそれぞれの単語間の関係を把握するための, 強化学習に基づいた方法を提案する. 強化学習には Xiong によって提案された DeepPath[2]を利用し, エージェントにナレッジグラフ上の関係を解釈するためのパスの選択方法を学習させることで犯人推定を行う.

2. ナレッジグラフについて

チャレンジ 2018 では, 推理小説で描かれる様々な状況をできるだけ統一的な形式で処理可能にする為 に, 内容である＜主語・述語・目的語＞を＜先頭エンティティ(h)・関係(r)・末尾エンティティ(t)＞のトリプルで表記するナレッジグラフ形式で表現している. ただし今回のナレッジグラフでは, 場面間の時間経過を考慮するために, 内容を場面ごとの最小単位に分割したものに ID を付与し, 登場人物やその行動の関係を表現した. 図 1 にナレッジグラフのイメージを示す.

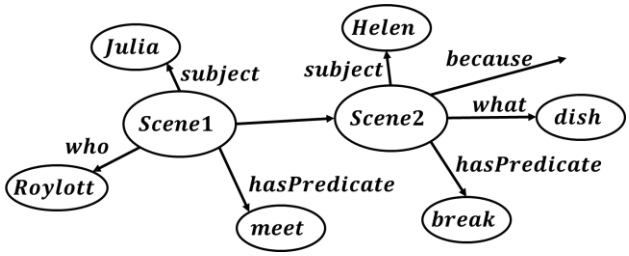

図 1 小説ナレッジグラフ

図 1 における場面間の関係（矢印で表される 遷移）の例を以下に記す.

- subject:場面間の記述において主語となる人やモノ
- hasPredicate: 場面の内容を表す述語
- 場面の詳細を表す目的語:who(誰), what(何)
- 場面間の関係:then, because など

†東京都市大学大学院 総合理工学研究科 情報専攻, Informatics, Graduate School of Integrative Science and Engineering, Tokyo City University Graduate School

3. 強化学習

強化学習(RL: Reinforcement Learning)とは, エージェントが, 与えられた目標を達成するために, 環境との相互作用によって, 得られる報酬を最大化するための行動規則を学習する枠組みである. 各ステップ t(t=0,1,2…)において, エージェントは環境の状態s_tを入力として受け取り, 方策πを用いた意思決定の後, 行動a_tを出力する. その後, 次のステップではエージェントはその行動の結果として報酬 R を受け取り, 状態s_{t+1}の観測値を受け取る. 本研究においてナレッジグラフが環境, 探偵がエージェントにあたる.

3.1 行動

エージェントは, 環境からエンティティのペア(e_{source}, e_{target})を与えられ, そのエンティティをつなぐような最も有益なパスの遷移を見つけようと行動する. ソースエンティティe_{source}からはじめて, エージェントは, ニューラルネットワークに基づいて出力される確率分布から関係を選択する. 選ばれた関係によってつながるエンティティのパスで次のエンティティへと遷移し, この遷移をターゲットエンティティe_{target}にたどり着くまで行う.

3.2 状態

エージェントが, 有益なパスの遷移を選択するために, ナレッジグラフ上の単語に意味が付与されている必要がある. 本研究では, 意味の学習ために TransE[3]や TransH などの translation-based embeddings を利用する.

3.2.1 Graph embedding

TransE では, 単語に割り当てられたベクトル関係を以下の式を用いて学習する.

$$f(h,r,t) = \|\mathbf{v}_h + \mathbf{v}_r - \mathbf{v}_t\| \quad (1)$$

$$\mathcal{L} = \sum_i [\tau + f(h_i, r_i, t_i) - f(h_i', r_i, t_i')]_+ \quad (2)$$

ここで, $\mathbf{v}_h$と$\mathbf{v}_r$, $\mathbf{v}_t$はそれぞれ先頭エンティティ, 関係, 末尾エンティティのベクトルである. また, $[x]_+$はヒンジ関数 $[x]_+ = \max(0, x)$であり, τはマージンである. $f(h,r,t)$は正例を表し, $f(h_i', r_i, t_i')$は負例を表す.

3.2.2 パス

これらナレッジグラフ上のエンティティの持つ埋め込みベクトルを, エージェントは状態として受け取る. あるステップ t においてエージェントの受け取る状態ベクトル以下のように定義する.

$$s_t = (e_{target} - e_t) \quad (3)$$

ここで, e_tは現在いるエンティティのノードの持つベクトル, e_{target}はターゲットエンティティの持つベクトルを指す. はじめの状態では, $e_t = e_{source}$である.

Copyright © 2020 by
The Institute of Electronics, Information and Communication Engineers and
Information Processing Society of Japan All rights reserved.

3.3 報酬

強化学習エージェントによるパスの有効性を高めるための報酬はいくつか考えられる. 本研究で利用する報酬は既存研究の DeepPath で利用されている大域正解報酬 r_{GLOBAL}, 効率性報酬 $r_{EFFICIENCY}$, 多様性報酬 $r_{DIVERSITY}$の 3 つを利用し, 以下に示す.

$$R_{total} = \lambda_1 r_{GLOBAL} + \lambda_2 r_{EFFICIENCY} + \lambda_3 r_{DIVERSITY} \quad (4)$$

ここで, $\lambda_1, \lambda_2, \lambda_3$はパラメータである. (4)式は以下に定義される 3 つの報酬関数の線形和によって, そのターンのエージェントに対する報酬R_{total}を算出する式である.

3.3.1 大域正解報酬

ナレッジグラフは一般的に非常にスケールが大きく, 今回のような状態獲得のためのパス遷移の場合, 正しい遷移より間違った遷移をたどる場合のほうが多い. そのため, 間違ったパスの遷移をおこなった場合に対しても報酬を設定する必要がある. エージェントのパスの遷移に対して与える大域正解報酬 r_{GLOBAL}を以下に示す.

$$r_{GLOBAL} = \begin{cases} +1, & if\ the\ path\ reaches\ e_{target} \\ -1, & otherwise \end{cases} \quad (5)$$

3.3.2 効率性報酬

単にパスの正解にのみを重要視すると非常に長いパスになってしまったり, 同じパスの巡回路に陥ってしまうことがある. パスの長さを制限するための効率性報酬 $r_{EFFICIENCY}$を以下に示す.

$$r_{EFFICIENCY} = \frac{1}{length(p)} \quad (6)$$

pはエージェントの選択したパス上の関係の遷移$(r_1 \rightarrow r_2 \rightarrow \cdots \rightarrow r_n)$を示す.

3.3.3 多様性報酬

本研究では, エージェントは正しいエンティティパスを関係を通して見つけられるように学習を行っていく. しかし, 学習によって得られるパスは似たような遷移となってしまうことが考えられる. これらのパスは正しいものだとしても, 冗長なパスとして避け, より多様なパスを見つけられるように多様性報酬 $r_{DIVERSITY}$を設定する.

$$r_{DIVERSITY} = -\frac{1}{|F|}\sum_{i=1}^{|F|} cos(\mathbf{p}, \mathbf{p}_i) \quad (7)$$

ここで, $\mathbf{p} = \sum_{i=1}^{n} \mathbf{r_i}$であり, $\mathbf{r_i}$は遷移上の関係の埋め込みベクトルである. F は記憶したパス数である. (7)式では, 現在のパスの遷移がすでに得られたパスの遷移との類似度を測るためにコサイン類似度を利用している.

3.4 方策勾配法

方策勾配法では, エージェント自身が行動確率を出力するための関数を持っており, その関数の変数群であるパラメータθを学習する. 具体的には, エージェントが確率的に選択した行動からθに対する目的関数の勾配を計算する. そしてその勾配から勾配降下法を用いて目的関数が最大になるように, θの値を更新することで, 方策そのものを直接的に変化させることができる.今回はこの方策勾配法に Williams の REINFORCE アルゴリズム[4]を利用する.

まず, ステップ t の状態 s において, パラメータθを持つエージェントが行動a_tを選択する確率的方策を$\pi(a_t|s)$とする. この方策によって, ネットワークから出力された確率分布に従い関係を選択し, その関係とつながるエンティティへと遷移する. これをターゲットエンティティにたどり着くか設定したパスの最大長に達するまでを 1 エピソードとする.それぞれのエピソードごとにニューラルネットワークを更新するために用いる, 最大化する目的関数$J(\theta)$の勾配を以下に記す.

$$\nabla_\theta J(\theta) = \nabla_\theta \sum_t \log \pi(a = r_t|s_t;\theta)\, R_{total} \quad (8)$$

REINFORCE アルゴリズムでは, θをこの勾配から勾配降下法を用いて更新している.

4. 学習方法

学習の流れを以下に記す.

① エンティティペアによる状態獲得
現在エージェント現在いるソースエンティティからターゲットエンティティまでの状態s_tを求める.

② 行動選択
得られた状態s_tから, パラメータθによるニューラルネットワークによって, それぞれの関係の行動確率を算出し, その確率に従い行動a_tを決定する. そして, 選択された関係とつながるエンティティへと遷移する.

③ 報酬の計算
遷移したエンティティによって, そこまでのパスを含めて評価する.

④ パラメータの更新
勾配降下法によって, パラメータの更新を行う.

①から⑥の流れを繰り返しながら,エージェントのパラメータθの最適化を図り学習する.

5. 今後の課題

提案手法による学習では, ニューラルネットワーク上で勾配消失が起きたり, 正しく学習ができていない. そもそものデータ上の単語の分布に偏りがあり, 出力された行動の確率的方策に偏りが出てしまっていることが考えられる. また, 現在の報酬関数では, パスの遷移を解釈するための有益な情報をすべて網羅していない可能性もあり, 報酬関数のさらなる見直しも必要であると考えられる. より正しいパラメータ設定とともに実験を繰り返したい.

参考文献

[1] 川村 隆浩, 江上 周作, 松下 京群, “第 1 回ナレッジグラフ推論チャレンジ 2018 開催報告~説明性のある人工知能システムを目指して~”, 人工知能学会研究会報告, Vol.34, No.3 (2019).

[2] Wenhan Xiong, Tjien Hoang, William Yang Wang “DeepPath: A Reinforcement Learning Method for Knowledge Graph Reasoning”, arXiv:1707.06690v3 [cs.CL] (2018).

[3] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran “Translating Embeddings for Modeling Multi-relational Data”, In Proceedings of the 26th International Conference on Neural Information Processing Systems(NIPS’13) (2013).

[4] Ronald Williams “Simple statistical gradient following algorithms for connectionist reinforcement learning”, In Machine Learning 8, 229-256 (1992).

Copyright © 2020 by
The Institute of Electronics, Information and Communication Engineers and
Information Processing Society of Japan All rights reserved.