

意味表現 N3 (Non-loss Normalized Network) と グラフマッチングによる機械読解エンジン

下村 光彦 弘中 智彦 井出 竜鳳 大洞 日音 倉田 伊織 永田 基樹 森 遼太

株式会社 pluszero

{shimomura.mitsuhiko, hironaka.tomohiko, ide.ryuhoh, daido.hinari, kurata.iori, nagata.motoki, mori.ryota}@plus-zero.co.jp

1 はじめに

機械読解では、質問に対して与えられた文章や所有している知識データを参照し、最も適切だと思われる回答を行う。しかしながら、通常の自然言語文では、修飾や倒置などによりその文構造は容易に変化するため、質問文で言及されている内容を本文中からそのまま参照することは非常に困難である。そのような表層的な違いを取り除き、単語間の本質的な関係性に注目した意味表現を目的とする中間形式として、自然言語のグラフ表現である N3 (Non-loss Normalized Network) を別稿 [1] で報告している。本論文では、この N3 を用いたグラフマッチング、具体的には本文と質問文それぞれを N3 に変換し、そのグラフ間の類似度を評価することにより機械読解を行う手法を提案する。

1.1 質問文から N3 への変換

平叙文と N3 の対応関係は別稿 [1] で報告しているが、冒頭で述べたグラフマッチングを行うには質問文についても N3 に変換する必要がある。そこで、質問文については、その質問対象を $\langle X \rangle$ とおいた平叙文に直した上で N3 に変換している。例えば、What did Tom eat for breakfast? という文章については Tom did eat $\langle X \rangle$ for breakfast. に直してから N3 へ変換している。

1.2 回答候補の絞り込み

質問文によっては、本文から回答の候補となるものを容易に絞り込むことができる場合がある。例えば What color ...? という質問であれば、本文中から色に関するノードだけを取り出せば良い。このように、与えられたクラスの下位語を答えさせる表現の他、When や Where などの時間場所表現なども、同様の絞り込みを行うことができる。このような、質問文による回答候補の絞り込みを取り入れたニューラルネットワークによって、SQuAD1.1[2] の exact match で 70.8 が出ることが知られている [3]。絞り込みは必ずしも行える

とは限らず、絞り込めない場合は本文に対応する N3 の全ノードを回答の候補とする。

1.3 各候補のスコア

1.2 節で抽出した各回答候補を $\langle Y \rangle$ とすると、 $\langle X \rangle$ を中心とする質問文の部分グラフと $\langle Y \rangle$ を中心とする本文の部分グラフの類似度を計算し、類似度が最大となるような $\langle Y \rangle$ をこの質問に対する回答とすることができる。次節以降でこの類似度計算の具体的な方法を説明する。

1.4 TFS

N3 による表現を一度木構造に変換し、木同士の比較演算により N3 の類似度を算出する。この木構造を Tree For Similarity (TFS) と呼ぶ。N3 から TFS への変換は次の手順で行われる。

1. $\langle X \rangle$ や $\langle Y \rangle$ といった N3 上で中心となるノードを、TFS 上の根とする。
2. TFS の各葉 L について、L に対応する N3 上のノードに隣接するリレーションのうち、TFS に未追加のものを L の子として追加する (N3 上のリレーションも TFS ではノードとして扱う)。
3. 2 で追加した N3 上の各リレーション R について、N3 上で R に隣接するノードのうち、TFS に未追加の全ノードを、R に対応する TFS 上のノードの子として追加する。
4. 2,3 の手順を、N3 の全てのノードが TFS に追加されるまで繰り返す。
5. 最後に、根のインデックスを 0 とし、そこから幅優先探索で各ノードにインデックスを振る (同じ深さであればインデックスを振る順番は任意である)。

最後のインデックスの振り方から、インデックスの大小関係とノードの深さの大小関係は等号付きで一致する。

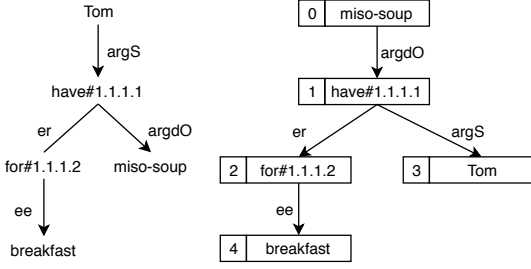


図 1: Tom had miso-soup for breakfast. に対応する N3 (左) と “miso-soup” を根としたときの TFS (右)

1.5 TFS を用いた類似度計算

実際の類似度計算アルゴリズムを与える前に、いくつか記号を導入する。

1.5.1 準備

まず、以降で繰り返し用いる記号を定義する。集合 A, B に対して $\text{Match}(A, B)$ を、次のような性質を満たす二項関係 $m \subset A \times B$ 全体の集合として定める。任意の $m \in \text{Match}(A, B)$ について

$$(a_1, b_1) \in m \wedge (a_2, b_2) \in m \Rightarrow (a_1 = a_2 \Leftrightarrow b_1 = b_2)$$

次に、1つの TFS Z について次の記号を定義する。ただし、 z は Z のノードとし、 z_0 は Z の根を表す。

- $N_Z :=$ TFS Z 上の全ノードからなる集合
- $\text{Path}_Z(z) := z_0$ から z までの単純路上の全ノードからなる集合
- $T_Z(z) := z$ の Z 上の親ノード
- $C_Z(z) := \{z' \mid T_Z(z') = z\}$

最後に、2つの TFS X, Y について各記号を次のように定義する。ただし、 $x_i \in N_X, y_j \in N_Y$ とする。

- $M_j^i := \text{Match}(C_X(x_i), C_Y(y_j))$
- $s_j^i := x_i$ と y_j に対応する単語間の類似度 (1.5.3 項参照)

1.5.2 アラインメント

TFS 間の類似度を定義するにあたって、アラインメントという概念を導入する。1.5.1 項で使用した X, Y を再び使用する。 \mathfrak{A}_{XY} を次の様な二項関係 $A \subset N_X \times N_Y$ 全体の集合として定義する。

任意の $A \in \mathfrak{A}_{XY}$ について

1. $A \in \text{Match}(N_X, N_Y)$
2. $(x_i, y_j) \in A \wedge (x, y) \in A \Rightarrow (x \in \text{Path}_X(x_i) \Leftrightarrow y \in \text{Path}_Y(y_j))$

以下、上記の 1,2 全てを満たす二項関係 A をアラインメントと呼ぶことにする。

このように定義したアラインメント A に対し、そのスコアを次のように定める。

$$\text{score}(A) := \sum_{(x_i, y_j) \in A} s_j^i$$

最後に、TFS X, Y の類似度 R_{XY} を次式で定義する。

$$R_{XY} := \max_{A \in \mathfrak{A}_{XY}} \text{score}(A)$$

1.5.3 類似度計算アルゴリズム

以下、質問文に対応する TFS を X 、本文に対応する TFS を Y とし、また $x_i \in N_X, y_j \in N_Y$ とする。

まず、TFS 上の各ノードに対応する単語をもとに s_j^i を計算する。本論文では Word2Vec[4] によりノード間類似度の計算を行なった。その際、モデルには Google の学習済みモデルを利用した。ただし、いま TFS 間の根同士がアラインメントに含まれる場合の類似度を計算するため、任意の i, j について $s_0^i = s_j^0 = 0$ とし、他のノードとのペアが含まれないようにした。

次に、TFS X, Y について階層の深いノードから順に、根に至るまで次式で与えられる r_j^i を更新する。各 i, j について、

$$r1_j^i := s_j^i + \max_{m \in M_j^i} \left\{ \sum_{(i_m, j_m) \in m} r_{j_m}^{i_m} \right\}$$

$$r2_j^i := \max_{y_j \in C_Y(y_j)} r_j^i$$

$$r3_j^i := \max_{x_i \in C_X(x_i)} r_j^i$$

なる $r1_j^i, r2_j^i, r3_j^i$ を用いて

$$r_j^i := \max \{r1_j^i, r2_j^i, r3_j^i\}$$

を計算する。ただし、 $m, C_Y(y_j), C_X(x_i)$ が \emptyset のときは、各部分に対応するスコアを 0 とする。先述の通り、インデックスの大小関係とノードの深さの大小関係は一致するから、 i, j が大きい方から順に r_j^i を計算すれば良い。最終的に r_0^0 が TFS 間の類似度 R_{XY} に一致することが確認できている。

以下、類似度計算の例を示す。

(本文) Tom had miso-soup for breakfast.

(問題) What did Tom eat for breakfast?

とし、 $\langle Y \rangle$ が “miso-soup” である場合について考える。対応する TFS を図 2 に示す。このとき、 s_j^i と r_j^i は表 1 のようになる。

表 1: 図 2 のときの s_j^i/r_j^i の値

| s_j^i/r_j^i | y_0 | y_1 | y_2 | y_3 | y_4 |
|---------------|-------------|-------------|-------------|-------------|-------------|
| x_0 | 0.00 / 3.65 | 0.00 / 3.65 | 0.00 / 2.00 | 0.00 / 1.00 | 0.00 / 1.00 |
| x_1 | 0.00 / 3.65 | 0.65 / 3.65 | 0.00 / 2.00 | 0.00 / 1.00 | 0.56 / 1.00 |
| x_2 | 0.00 / 2.00 | 0.00 / 2.00 | 1.00 / 2.00 | 0.00 / 0.00 | 0.00 / 1.00 |
| x_3 | 0.00 / 1.00 | 0.00 / 1.00 | 0.00 / 0.00 | 1.00 / 1.00 | 0.00 / 0.00 |
| x_4 | 0.00 / 0.84 | 0.00 / 0.84 | 0.00 / 0.50 | 0.00 / 0.00 | 0.00 / 0.50 |
| x_5 | 0.00 / 1.00 | 0.55 / 1.00 | 0.00 / 1.00 | 0.00 / 0.00 | 1.00 / 1.00 |
| x_6 | 0.00 / 0.84 | 0.84 / 0.84 | 0.00 / 0.50 | 0.00 / 0.00 | 0.50 / 0.50 |

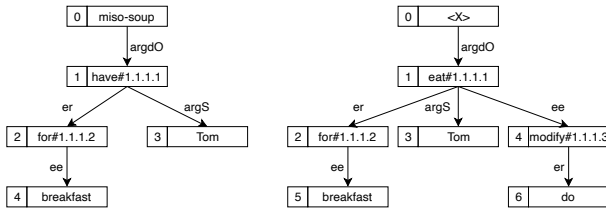


図 2: “miso-soup” を $\langle Y \rangle$ としたときのマッチングに使用した両 TFS

2 評価実験及び考察

1章で提案したアルゴリズムを用いることで、実際に機械読解が行えることを確認する。機械読解のデータセットには SQuAD[2] を使用した。まず本文と質問文のそれぞれを N3 に変換し、それらに対して 1章に示した手順で処理を行い、スコアが高い順に回答した。一部の文章については、構文解析が上手くいかず、N3 に変換できないものが存在した。そのような文章については、手動で N3 に変換してからアルゴリズムを適用した。

2.1 結果

本研究では一部の N3 を手動で構築する都合上、多くの文章では実験していない。本節では読解が可能であった文章および不可能であった文章について述べる。まず、読解できた文章についてその特徴と例文を以下に示す。

- 回答の候補が 1 つに絞れるもの
 - 1.2 節で述べた方法のみで候補が 1 つに絞れるような、単純な問題は回答することができた。
- グラフの形状がほぼ一致するもの
 - (本文) Computational complexity theory focuses on classifying computational problems according to their inherent difficulty.
 - (問題) By what main attribute are computational problems classified utilizing computational complexity theory?

これらに対応する TFS を図 3, 4 に示す。両グラフとも “classify” 以下の構造が非常に類似していることがわかる。このような場合は、本論文で提案したグラフマッチングのみで回答することが可能である。

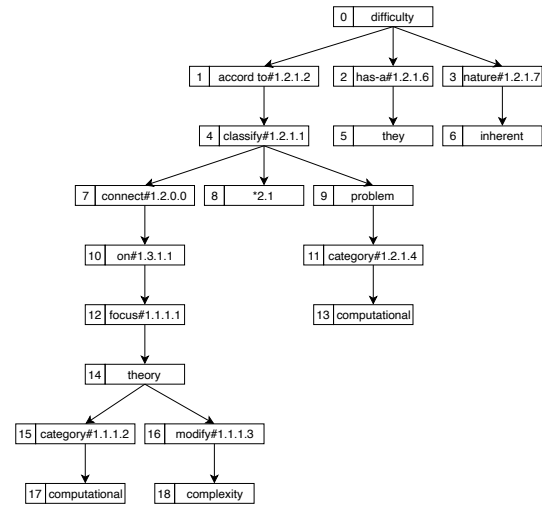


図 3: 本文で “difficulty” を根としたときの TFS

次に、読解できなかった文章についてその特徴と例文を以下に示す。

- 複合語の類似度評価が必要なもの
 - (本文) The American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference (NFC) champion Carolina Panthers 24–10 to earn their third Super Bowl title.
 - (問題) What is the AFC short for?

この例では “be short for” が略称を意味する熟語であることを認識し、“abbreviation” との類似度評価が必要である。これらに対応する部分 N3 を図 5 に示す。

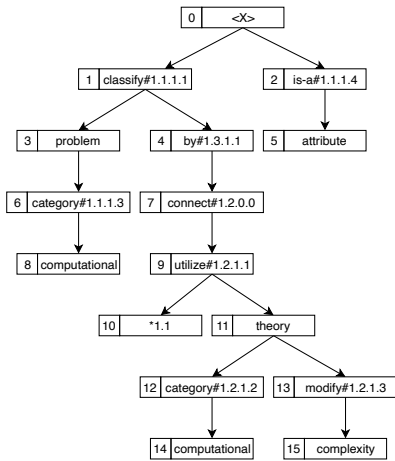


図 4: 質問文で “<X>” を根としたときの TFS

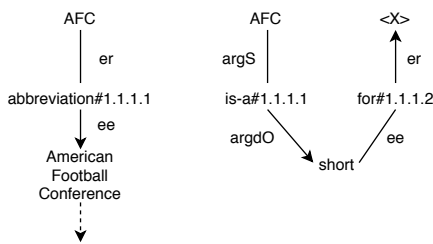


図 5: “abbreviation” と “be short for” それぞれに対応する N3

- 複合語に加えグラフの構造が大きく異なるもの
 - (本文) 同上
 - (問題) Who won Super Bowl 50?

この例では、まず Super Bowl が大会であることを認識し、“win Super Bowl” と “earn Super Bowl title” とが同じ意味である事を理解する必要がある。

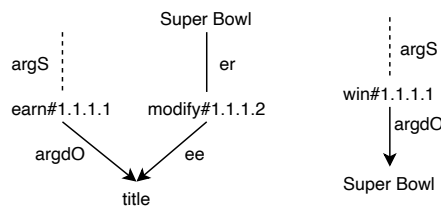


図 6: “earn title” と “win” それぞれに対応する N3

2.2 考察

本節ではより高度な読解を行うための手法について考察する。主には 2.1 節で述べた読解不可能な文章に関して改善案を考案する。まず、熟語の類似度評価、より一般に Multi-Word Expression (MWE) の類似度評価については、何らかのコーパスを用いて計算する方法が考えられるが、現状あまり有効な手法は報告され

ていない。ドメインを非常に限定した場合についての研究 [5] が存在する程度である。一方、compositional MWE (構成する単語から意味を推測できる MWE) については、その構成単語について類似度評価を行うことで、MWE の有効な類似度評価が行えるのではないかと考えられる。また、グラフ構造が異なるものとして挙げた例についても、“win” と “earn title” の類似という意味で先述の MWE の議論に帰着することができる。さらに、本論文には載せていないが、機械読解の問題によっては時制や量子子が必要になるものもあり、それらの解決も必要である。

3 おわりに

通常自然言語文は、その表層的な違いにより質問されている箇所を適切に参照することが困難であった。本研究では、自然言語のグラフ表現 N3 を用いると、グラフ間類似度の計算により機械読解が行えることを示し、その具体的な類似度計算アルゴリズムを提案した。さらに、MWE の類似度を適切に評価することができれば、より多くの読解問題にも適用できる可能性を示した。

本研究では N3 の表層的なラベルやグラフ構造のみを使用し、高度な推論は行っていない。本来 N3 は、時制や量子子を導入することで、より高度な推論を行うことを目的としている。そのような推論が行えるアルゴリズムを開発することが、今後の課題である。

参考文献

- [1] 弘中智彦, 井出竜鳳, 下村光彦, 大洞日音, 倉田伊織, 永田基樹, 森遼太. “意味表現 N3: Non-loss Normalized Network”. 言語処理学会 2020 投稿済み, 2020.
- [2] Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. “Squad: 100, 000+ questions for machine comprehension of text.” Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016.
- [3] Weissenborn, D., Wiese, G., and Seiffe, L. “Making neural QA as simple as possible but not simpler.” Proceedings of the 21st Conference on Computational Natural Language Learning, 2017.
- [4] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. “Distributed Representations of Words and Phrases and their Compositionality.” In Proceedings of NIPS, 2013.
- [5] Spasić, I., Corcoran, P., Gagarin, A., and Buerki, A. “Head to Head: Semantic Similarity of Multi-Word Terms.” IEEE Access 6, 2018.