記憶装置付きニューラルネットワークモデルによる構造化知識と演算処理を用いた質問応答

村山友理　小林一郎
お茶の水女子大学
{murayama.yuri,koba}@is.ocha.ac.jp

概要

3 つの DNC モデル，vanilla DNC, rsDNC, DNC-DMS に知識と演算のためのアーキテクチャを新たに組み入れ，構造化知識に対する演算処理を含んだ質問文について正しい答えを生成する能力を向上させることを目指す．rsDNC, DNC-DMS をベースとした提案モデルはそれぞれ GEO データセットにおいて平均 top-1 accuracy，平均 top-10 accuracy で最も良い結果を達成した．さらに，rsDNC をベースとした提案手法は拡張した GEO データセットにおいて平均 top-1 accuracy と平均 top-10 accuracy の両方で他のモデルを上回った．

1　はじめに

近年，Transformer [1] などのディープニューラルネットワークはコンピュータビジョンや自然言語処理といったさまざまなタスクの複雑なパターンマッチングにおいて顕著な発展を遂げてきた．しかし，Transformer には長い文脈情報を固定長の系列にエンコードするせいで context fragmentation problem [2] があり，この問題を解くために過去の情報をメモリ内にキャッシュしておくさまざまな手法 [2, 3, 4] が提案されてきたが，Transformer をベースとしたモデルは，グラフや木などのデータ構造の表現や変数の使用，長い系列に対する表現の操作といった抽象的な処理を行う能力には限界があるとされてきた．Neural Turing Machine [5] や Differentiable Neural Computer (DNC) [6] は外部に読み書き可能なメモリを持つことで，構造化データ上のアルゴリズムタスクを解き，変数の表現や，長い系列の学習を可能にした．本研究では，DNC と，さらに DNC を改良した rsDNC [7] と DNC-DMS [8] に対して，質問応答タスクにおいて重要な要素である知識利用と演算処理を新たに組み入れることを試み，構造化知識に対する演算処理を含んだ質問文について正しい答えを生成する能力を向上させることを目指す．

2　関連研究

Differentiable Neural Computer (DNC) [6] は外部にメモリ行列 $M \in \mathbb{R}^{N\times W}$ を持つニューラルネットワークである．メモリ行列 M の N 個の番地に対し，どの番地について主に読み出す，または書き込むかを表す重みを定義するのに attention mechanism が用いられる．

読み出し操作では，read vector $\boldsymbol{r}$ はメモリ M に read weighting $\boldsymbol{w}^r$ をかけた，メモリ番地に対する重み付き和として計算される：

$$\boldsymbol{r} = \sum_{i=1}^{N} M[i,\cdot]\boldsymbol{w}^r[i]$$

ここで，‘·’ は $j = 1, ..., W$ を表す．

書き込み操作では，メモリ M は write weighting $\boldsymbol{w}^w$ を用いてまず erase vector $\boldsymbol{e}$ により不要な番地が消去され，write vector $\boldsymbol{v}$ を足すことで更新される：

$$M[i,j] \leftarrow M[i,j](1 - \boldsymbol{w}^w[i]\boldsymbol{e}[j]) + \boldsymbol{w}^w[i]\boldsymbol{v}[j]$$

重みは内容に基づく番地付け，一時的なメモリのリンク付け，動的メモリ割り当て，の 3 つの attention mechanism によって定義される．DNC を改良したモデルとして，QA タスクに特化した robust and scalable DNC (rsDNC) [7] や DNC に対して 3 つの改良 (i.e. de-allocation mechanisms, masked content based addressing, sharpness enhancement) を行なった DNC-DMS [8] などが提案されている．

3　提案手法

3 つのモデル，DNC [6], rsDNC [7], DNC-DMS [8] に新たに構造化知識を保存するためのメモリアーキテクチャと，単純な算術演算と論理演算を行うためのプロセッサアーキテクチャを追加する．図 1 は

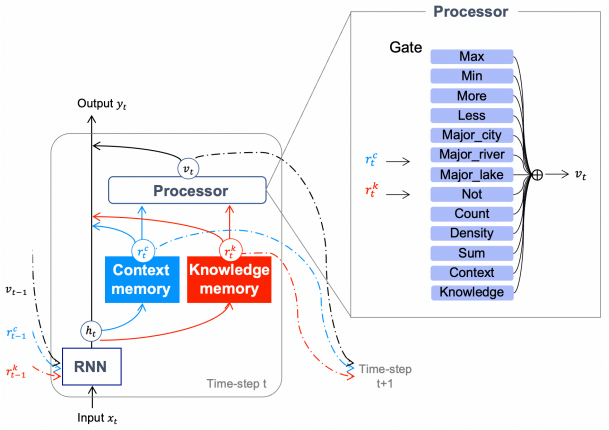

図 1 知識メモリとプロセッサを持つ提案モデルの全体図.

DNC に基づく知識メモリとプロセッサを持つ提案モデルの全体図を示す．各タイムステップ t で行う処理は以下の通りである：

1. コントローラ (RNN) は入力 $\boldsymbol{x}_t$ と，前タイムステップで文脈メモリ $M^c_{t-1} \in \mathbb{R}^{N\times W}$ から読み出した R 個のベクトルのセット $\boldsymbol{r}^c_{t-1} = [\boldsymbol{r}^{c,1}_{t-1}; ...; \boldsymbol{r}^{c,R}_{t-1}]$ ($\boldsymbol{r}^c_{t-1}$ は $\boldsymbol{r}^{c,1}_{t-1}, ..., \boldsymbol{r}^{c,R}_{t-1}$ の結合) に加えて，前タイムステップで知識メモリ $M^k_{t-1} \in \mathbb{R}^{N\times W}$ から読み出した R 個のベクトルのセット $\boldsymbol{r}^k_{t-1} = [\boldsymbol{r}^{k,1}_{t-1}; ...; \boldsymbol{r}^{k,R}_{t-1}]$，そして前タイムステップのプロセッサの演算結果のベクトル $\boldsymbol{v}_{t-1}$ を受け取る．それから隠れベクトル $\boldsymbol{h}_t$ を出力する．
2. $\boldsymbol{h}_t$ の線形変換により，出力 $\boldsymbol{v}_t = W_y\boldsymbol{h}_t$ と，現タイムステップにおける文脈メモリを制御するためのパラメータを格納したベクトル $\boldsymbol{\xi}_t = W_\xi\boldsymbol{h}_t$，現タイムステップにおける知識メモリを制御するためのベクトル $\boldsymbol{\zeta}_t = W_\zeta\boldsymbol{h}_t$，そして現タイムステップのプロセッサに用いられるゲートベクトル $\boldsymbol{g}_t = W_g\boldsymbol{h}_t$ を得る．
3. $\boldsymbol{\xi}_t$ によって文脈メモリへの書き込みが行われ，メモリの状態が更新される．知識メモリへの書き込みは行われない．
4. プロセッサでの演算処理は $\boldsymbol{g}_t$ と，現タイムステップで文脈メモリから読み出したベクトルを結合した $\boldsymbol{r}^c_t$，現タイムステップで知識メモリから読み出したベクトルを結合した $\boldsymbol{r}^k_t$ を用いて行われる．
5. 最後に，$\boldsymbol{r}^c_t$ と W^c_r をかけて得られるベクトルと，$\boldsymbol{r}^k_t$ と W^k_r をかけて得られるベクトル，及び現タイムステップのプロセッサからのベクトル $\boldsymbol{v}_t$ と W_v をかけて得られるベクトルに $\boldsymbol{v}_t$ を足し，出力 $\boldsymbol{y}_t$ を計算する．

$$\boldsymbol{y}_t = \boldsymbol{v}_t + W^c_r\boldsymbol{r}^c_t + W^k_r\boldsymbol{r}^k_t + W_v\boldsymbol{v}_t$$

read vector $\boldsymbol{r}^c_t$ と $\boldsymbol{r}^k_t$，value vector $\boldsymbol{v}_t$ は次タイムステップの RNN への入力に追加される．

以上の処理を繰り返すことにより，二つのメモリへの読み書き操作とプロセッサ操作を行う．

3.1 知識メモリ構築

知識メモリは知識ベース (KB) を用いて構築される．KB のファクトは Resource Description Framework (RDF)[1] 形式の三つ組（主語，述語，目的語）で表現される．例えば，「日本の首都は東京である．」は三つ組 (日本，首都，東京) で表される．DNC のモデルに三つ組の内どれか 2 つを与え，残りの 1 つを返すように学習させることで KB ファクトを学習する．例えば，入力が “日本”，“首都” のとき，出力は “東京” である．モデルは KB のすべてのトリプルを用いて学習し，KB 全体を保存したメモリユニットを作成する．そして，事前学習したメモリユニットを提案モデルの知識メモリユニットとして利用する．

3.2 プロセッサ処理

プロセッサユニットにおいて単純な算術演算と論理演算を行うために，13 の演算を設定した：`Max`, `Min`, `More`, `Less`, `Major_city`, `Major_river`, `Major_lake`, `Not`, `Count`, `Density`, `Sum`, `Context`, `Knowledge`. 文脈メモリからの read vector $\boldsymbol{r}^c$ と知識メモリからの read vector $\boldsymbol{r}^k$ はオリジナルの語彙トークンのリスト，「文脈リスト」と「知識リスト」にそれぞれ変換される．

Superlatives ：`Max` 演算は知識リストを受け取り，その最大値を返す．`Min` 演算は同様に知識リストを受け取り，その最小値を返す．

Comparatives ：`More` 演算は文脈リストと知識リストの両方を受け取り，知識リストの最初の数より大きい数を文脈リスト中から返す．`Less` 演算は同様に文脈リストと知識リストの両方を受け取り，知識リストの最初の数より小さい数を文脈リスト中から返す．`Major_city`, `Major_river`, `Major_lake` 演算は知識リストを受け取り，それぞれ “150,000”, “750”, “5,000” より大きい数を返す．

1) https://www.w3.org/TR/rdf11-primer/

表 1 知識ベースの一部.

type	alabama	state
capital	alabama	montgomery
population	alabama	3894.0e+3

Negation ：`Not` 演算は文脈リストと知識リストの両方を受け取り，文脈リストから知識リストを引いた差を返す.

Calculation ：`Count` 演算は知識リストを受け取り，その要素数を返す. `Density` 演算は文脈リストと知識リストの両方を受け取り，要素毎に知識リストで割られた文脈リストを返す. `Sum` 演算は知識リストを受け取り，その要素の和を返す.

No operation ：`Context` 演算は文脈リストを受け取り，それ自身を返し，`Knowledge` 演算は知識リストを受け取り，同様にそれ自身を返す. これらの 2 つの演算は上記の演算がどれも適当でない場合のために用意した.

13 の演算の出力は gate vector $\boldsymbol{g}$ のソフトマックス出力と結合し，value vector $\boldsymbol{v}$ を構築する.

4 実験

4.1 データセット

GEO データセット [9][2] はアメリカの地理に関する 880 の質問文と Prolog 形式のファクトのデータベースを含む. 質問文の語彙サイズは 280 である. 質問文は superlatives, comparatives, negation や count, density, sum といった calculation を含む. GEO データセットには質問文に対する答えが含まれていなかったため，答えを人手でアノテーションを行った. 答えはデータベースからの地理エンティティのリストであり，長さは 0 から 386 エンティティである. 例えば，“Which rivers flow through Alaska ?” という質問は答えがなく，別の質問 “Give me the cities in the U.S. ?” は 386 エンティティというかなり多くの答えを持つ. 600 サンプルを学習に，280 サンプルをテストに用いた.

GEO のデータベースから RDF 形式の三つ組を作成し，表 1 に示す. 三つ組 (`type`, `alabama`, `state`) は `alabama` の `type` は `state` であると意味する. `type` リレーションに対するオブジェクトエンティティとして，5 つのエンティティ：`state`, `city`, `river`, `mountain`, `lake` を我々の KB に追加した. この KB は地理ドメイン内の 11 リレーション，1,275 エンティティ，2,250 トリプルを含む.

さらに, 新たな 500 QA ペアを人手で作成し GEO データセットを拡張した. この拡張したデータセットを “GEO 1380” と呼ぶ. 1,000 サンプルを学習に，380 サンプルをテストに用いた.

2） https://cs.stanford.edu/ pliang/software/

表 2 GEO データセットと GEO 1380 における平均 Acc@k.

	GEO		GEO 1380	
	Acc@1	Acc@10	Acc@1	Acc@10
DNC [6]	20.93	44.98	24.37	51.16
rsDNC [7]	20.76	44.86	26.04	53.41
DNC-DMS [8]	20.20	44.46	25.37	52.53
DNC+KM	19.53	43.58	25.39	52.82
rsDNC+KM	20.69	43.98	**26.08**	**53.87**
DNC-DMS+KM	20.88	**45.40**	25.10	52.23
DNC+KM+P	20.03	43.82	23.76	51.46
rsDNC+KM+P	**21.20**	45.16	25.79	53.28
DNC-DMS+KM+P	19.51	43.58	22.82	50.69

4.2 結果

表 2 に GEO データセットと GEO 1380 における全てのモデルのテスト 3 実行分の平均 top-1 accuracy (Acc@1) と top-10 accuracy (Acc@10) を示す. “+KM” と “+P” はそれぞれ提案手法である知識メモリユニット (KM) の追加，プロセッサユニット (P) の追加を表す.

GEO データセットにおいて，rsDNC+KM+P は Acc@1 で最も良い結果を達成し， DNC-DMS+KM は Acc@10 で最も高いスコアを得た. GEO 1380 データセットでは，rsDNC+KM は他のモデルを上回った. QA タスクに特化した rsDNC を基にしたモデルは良い結果になる傾向がある. 提案モデルのほとんどはオリジナルのモデルより低いが，提案手法のポイントは異なるハイパーパラメータだと上回ることがあるため，アーキテクチャによるものだとは考えていない. すべてのスコアが GEO 1380 で上がったため，より大きなデータセットはモデルの精度向上に有効である.

表 3 に，GEO 1380 データセットにおける全てのモデルの各質問タイプのテスト 3 実行分の平均 Acc@10 を示す.

Simple ：1Hop は “What states border Texas?” のように質問文が 1 つのリレーションを含むことを意味する. 例えば，“What states border states that border states that border states that border Texas?” は 4 つのリ

表 3 GEO 1380 データセットにおける各質問タイプの平均 Acc@10.

		DNC	rsDNC	DNC-dms	DNC +KM	rsDNC +KM	DNC-dms +KM	DNC +KM+P	rsDNC +KM+P	DNC-dms +KM+P
Simple	1Hop	74.83	75.98	76.29	77.03	**79.23**	76.93	78.46	77.85	74.59
	2Hop	23.60	25.67	**26.53**	23.77	24.29	23.94	21.96	24.29	22.74
	4Hop	46.43	60.71	59.52	**66.67**	51.19	**66.67**	50.00	52.38	47.62
Superlatives	Argmax	67.44	67.83	68.60	68.99	68.22	67.83	67.44	**69.38**	67.05
	Argmax-1Hop	85.28	85.28	85.06	**86.15**	85.93	85.50	81.82	85.71	84.42
	Argmax-2Hop	75.85	**78.91**	76.19	75.51	76.53	75.85	72.45	77.21	77.21
	Argmin	79.63	82.10	79.01	77.16	82.10	79.01	75.93	**83.33**	80.86
	Argmin-1Hop	88.44	89.12	89.80	**92.52**	88.44	89.12	85.71	90.48	83.67
	Argmin-2Hop	43.79	50.33	**56.21**	50.98	47.71	48.37	41.18	54.90	32.68
Comparatives	Major	70.62	84.18	77.40	80.23	**86.44**	68.36	79.66	77.40	72.88
	Major-1Hop	42.69	**49.26**	40.72	42.86	46.96	40.23	36.95	45.65	39.90
	Major-2Hop	21.43	22.79	21.43	20.07	21.77	**23.13**	22.45	20.75	21.09
Negation	Not-1Hop	52.27	63.64	60.61	60.86	**65.15**	56.31	54.55	60.35	56.06
Calculation	Count	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	Count-1Hop	96.00	96.00	85.33	**97.33**	92.00	90.67	88.00	94.67	93.33
	Count-2Hop	**100.0**	**100.0**	66.67	**100.0**	**100.0**	83.33	**100.0**	**100.0**	**100.0**
	Density	8.11	7.68	7.62	8.59	**9.32**	8.05	8.35	8.23	8.41
	Sum-1Hop	81.25	81.25	81.25	81.25	81.25	81.25	81.25	81.25	81.25
	All	51.16	53.41	52.53	52.82	**53.87**	52.23	51.46	53.28	50.69

レーションを含むため，この例は 4Hop である．1Hop は GEO と GEO 1380 の両方の 30%以上を占めるため，このタイプで高いスコアを得ることが全体の良い結果につながる．4Hop のポイントが 2Hop のポイントより高いのは，4Hop の方が 2Hop より難しいにも関わらず奇妙に思えるかもしれないが，これは 4Hop のサンプル数が少ないせいだと考えられ，multi-hop のサンプルを拡張する必要がある．

Superlatives ： Argmax は “What is the largest state?” のように質問文が “largest”, “highest”, “longest” などといった単語を含むことを意味する．Argmax と Argmin において，rsDNC+KM+P が最も高いスコアを得た．superlatives の他のタイプでは，rsDNC+KM+P はトップスコアに届かなかったが，それでもベストなモデルと比べて遜色ない．

Comparatives ： このタイプは “What states high point are higher than that of Colorado?” のように 3 つの表現：More, Less, Major を扱う．ホップ数が増えると，スコアは下がった．

Negation ： Not は “What state has no rivers?” のように質問文が否定表現を含むことを意味する．我々の rsDNC+KM は他のモデルを上回った．

Calculation ： このタイプは “How many states are in the USA?” のように 3 つの演算：Count, Density, Sum を扱う．Count と Sum のポイントは他のタイプと比べて比較的高い一方で，Density の結果はとても低い．

4.3 課題

質問タイプ数のバランスを取るためにデータセットをより拡張する必要がある．また，GEO データセットのサイズは僅か 880 であり，拡張を行なってもまだ 1380 であり，ニューラルネットワークの学習には小さすぎるため，WikiTableQuestions [10] のような大規模データセットを用いて学習することが望まれる．

5 おわりに

3 つの DNC モデル，vanilla DNC, rsDNC, DNC-DMS に知識メモリとプロセッサを追加し，実験を行い，背景知識や単純な算術演算と論理演算を必要とする質問応答タスクに対して効果を分析した．提案した rsDNC+KM+P と DNC-DMS+KM はそれぞれ GEO データセットにおいて平均 Acc@1 と平均 Acc@10 で最も良い結果を達成した．さらに，提案モデル rsDNC+KM は GEO 1380 データセットにおいて平均 Acc@1 と平均 Acc@10 の両方で他のモデルを上回った．今後の課題では，提案モデルに順次実行，条件分岐，反復といった制御命令を処理するアーキテクチャを加えて改良したい．

謝辞

本研究は JSPS 科研費 JP20J23182 の助成を受けたものです.

参考文献

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, L ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, **Advances in Neural Information Processing Systems**, Vol. 30. Curran Associates, Inc., 2017.

[2] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, pp. 2978–2988, Florence, Italy, July 2019. Association for Computational Linguistics.

[3] Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, and Timothy P. Lillicrap. Compressive transformers for long-range sequence modelling. **CoRR**, Vol. abs/1911.05507, , 2019.

[4] Pedro Henrique Martins, Zita Marinho, and Andre Martins. ∞-former: Infinite memory transformer. In **Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 5468–5485, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[5] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines, 2014. cite arxiv:1410.5401.

[6] Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, Adrià Puigdomènech Badia, Karl Moritz Hermann, Yori Zwols, Georg Ostrovski, Adam Cain, Helen King, Christopher Summerfield, Phil Blunsom, Koray Kavukcuoglu, and Demis Hassabis. Hybrid computing using a neural network with dynamic external memory. **Nature**, Vol. 538, No. 7626, pp. 471–476, October 2016.

[7] Jörg Franke, Jan Niehues, and Alex Waibel. Robust and scalable differentiable neural computer for question answering. **CoRR**, Vol. abs/1807.02658, , 2018.

[8] Róbert Csordás and Jürgen Schmidhuber. Improving differentiable neural computers through memory masking, de-allocation, and link distribution sharpness control. **CoRR**, Vol. abs/1904.10278, , 2019.

[9] John M. Zelle and Raymond J. Mooney. Learning to parse database queries using inductive logic programming. In **Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 2**, AAAI'96, p. 1050–1055. AAAI Press, 1996.

[10] Panupong Pasupat and Percy Liang. Compositional semantic parsing on semi-structured tables. In **Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)**, pp. 1470–1480, Beijing, China, July 2015. Association for Computational Linguistics.

[11] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. **Neural computation**, Vol. 9, No. 8, pp. 1735–1780, 1997.

[12] T Tieleman and G Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. 2012.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)**, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[14] Mor Geva, Ankit Gupta, and Jonathan Berant. Injecting numerical reasoning skills into language models. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 946–958, Online, July 2020. Association for Computational Linguistics.

[15] M. Stone. Cross-validatory choice and assessment of statistical predictions. **Roy. Stat. Soc.**, Vol. 36, pp. 111–147, 1974.

A　実験設定

すべてのモデルのハイパーパラメータは主に [6] に基づく；隠れ層サイズ 256 の 1 層 LSTM [11]，バッチサイズ 2，学習率 1×10^{-4}，メモリ次元 256×64，読み出しヘッド数 4，書き込みヘッド数 1，モメンタム 0.9 の RMSProp オプティマイザ [12]．[7] に従い rsDNC [7] のドロップアウト率は 10%とした．HuggingFace[3] bert-base-uncased model を隠れ次元 768 の BERT [13] encoder に使用した．数字は [14] を参考に 1 桁ずつ分割した．5 分割交差検証 [15] で 5 エポックずつモデルを学習させた．ランダムな初期化の下で各モデルを 3 回実行し，平均の結果を報告する．

知識ベースを用いて知識メモリを構築するために，3 つのオリジナルモデルをメモリ次元 256×64 で，5 分割交差検証で 10 エポックずつ学習させた．他の設定は前述の通りである．DNC, rsDNC, DNC-DMS の top-10 accuracy はそれぞれ 78.90%, 78.39%, 79.63%だった．知識メモリ構築に用いる事前学習モデルと提案手法に用いる学習モデルは 同じ種類である．つまり，DNC を用いて学習した知識メモリが，DNC をベースとした提案モデルで使用される．

B　データセット

GEO データセット [9] と GEO 1380 の各質問タイプ数を表 4 に示す．Compound タイプは “How many states have a higher point than the highest point of the state with the largest capital city in the U.S.?” のように 4 つのタイプ：Superlatives, Comparatives, Negation, Calculation を複合した質問文を扱っており，さまざまな組み合わせが用意されている．

表 4　各質問タイプ数.

		GEO	GEO 1380
Simple	1Hop	94	145
	2Hop	10	11
	4Hop	1	1
Superlatives	Argmax	20	27
	Argmax-1Hop	43	49
	Argmax-2Hop	21	24
	Argmin	10	16
	Argmin-1Hop	13	15
	Argmin-2Hop	4	5
Comparatives	Major	3	5
	Major-1Hop	8	12
	Major-2Hop	2	3
Negation	Not-1Hop	2	3
Calculation	Count	8	11
	Count-1Hop	6	11
	Count-2Hop	1	1
	Density	3	4
	Sum-1Hop	2	2
Compound		30	35
	All	280	380

3）　https://github.com/huggingface/transformers