STILTs を適用した機械読解による Wikipedia 記事の構造化

石井 愛
日本ユニシス株式会社 総合技術研究所
ai.ishii@unisys.co.jp

1 はじめに

自然なコミュニケーションで問題解決に導いていく人に寄り添う AI を実現するためには，人の言葉の文脈を理解することや，人の常識や知識を共有することが必要とされる．より大規模で体系だった知識ベースの整備が望まれる中で，森羅プロジェクト[1]では, Wikipedia を機械可読な構造に変換する構造化を目指している．既存の知識ベースの一貫性の課題を解決するため，森羅では，あらかじめ定義された名前のオントロジーである「拡張固有表現」[2]（以降，ENE）に Wikipedia 記事を分類し，ENE に定義されている属性に対応する値を記事中から抽出する．森羅では，ENE に定義されている属性に対応する値を記事中から抽出する部分を「日本語 Wikipedia 構造化タスク」として，評価型ワークショップを開催し，同時に参加者のシステムによるリソース構築を進めている．

本稿では，日本語 Wikipedia 構造化タスクを，機械読解タスクとして定式化し，BERT[3]の事前学習モデルを用いて解くシステムにおいて，STILTs[4]と呼ばれる中間タスクを用いる手法を提案する．提案手法は，スパースなデータセットにおいて，限られた GPU リソース上で実行可能で，かつ機械読解の性能を向上させる効果的な学習を可能とする．評価実験では，中間タスクを用いない場合と比較し，提案手法の有効性を検証する．

2 日本語 Wikipedia 構造化タスク

日本語 Wikipedia 構造化タスクは，Wikipedia 記事のカテゴリごとに ENE に定義されている属性に対応する値を Wikipedia 記事中から抽出するタスクである．たとえば「企業名」カテゴリでは，属性として「正式名称」,「事業内容」など 34 種類の属性が ENE で定義されている[5]. あらかじめ各カテゴリに分類された Wikipedia 記事と，属性値の記事中の出現箇所をアノテーションした教師データが与えられ，それをもとに未アノテーションの記事を含むすべての記事から属性値を抽出する．そのうち非公開の 100 記事を用いて評価が行われる[6].

2019 年に開催された「森羅 2019-JP」では，記事数の多いカテゴリを中心とした JP-5 グループの，人名，市区町村名，企業名，化合物名，空港名の 5 カテゴリおよび，地名グループと組織名グループからなる 28 カテゴリが対象とされた．アノテーションデータは，JP-5 は各カテゴリ約 1000 記事，地名および組織名は各カテゴリ約 200 記事である. 属性値は，1 記事内に複数存在する場合や，一つも存在しない場合がある. JP-5 のアノテーションデータにおける，一つの記事内の属性値の個数を平均すると，最大の属性で 46.8，最小の属性で 0.03，カテゴリごとの中央値は 0.53～1.03 であり，属性によっては非常にスパースなデータである．

3 関連研究

日本語 Wikipedia 構造化タスクと直接関連が強い固有表現抽出において，Li ら[7, 8]は，BERT を用いた機械読解手法で他の手法よりも良い精度を達成している[9]．これまで開催された日本語 Wikipedia 構造化タスクにおいても，DrQA[10]を応用した機械読解システム[11]や RoBERTa[12]を用いた機械読解システム[13]が良い成績を収めている[1, 6]．評価型タスクの結果の分析では，属性値が単語ではなく文で記載されやすい「製造方法」などの属性の抽出，および訓練データが少ないカテゴリにおける属性値の抽出で精度が伸びないことが課題として示された．訓練データが少ないカテゴリにおいては，カテゴリ間で共通する属性が存在するため，[13]のシステムのようにカテゴリを横断して学習を行うことの有効性が示唆されている[6]. ただし，BERT や RoBERTa のような事前学習モデルを用いて多数あるカテゴリを横断して学習を行うには，大きな GPU リソースが必要となる．

表 1 機械読解用に変換したデータの例

p	サントペコア国際空港(サントペコアこくさいくうこう,Santo-Pekoa International Airport)とは、<a>バヌアツ,<a>エスピリトゥサント島にある国際空港。 …(中略)…滑走路の表面には珊瑚が使用されていた。1970 年代に民間用空港として開港している。 ¥n ¥n
q, A	所在地, [{"answer_end": 301, "answer_start": 291, "text": "エスピリトゥサント島"}] $(k = 1)$
q, A	近隣空港, [] $(k = 0)$

機械読解は，質問と，解答のための情報源となるテキスト（パッセージ）を入力として，解答となる情報をパッセージから抽出するタスクである．機械読解システムは，質問に答えるだけでなく，与えられたパッセージに解答となる情報が含まれていない場合にそれを判断する必要がある．解答できない問題を含む機械読解のデータセットである SQuAD 2.0[14]において，Retrospective Reader[15]は，パッセージに解答が含まれているかどうかを 2 値分類するタスクと，パッセージから解答を抽出する機械読解タスクに役割をわけ，別々のモデルで学習，予測した結果を統合することで, 良い精度を達成している．

BERT のような事前学習モデルの性能をさらに改善する方法の一つとして，最初に中間タスクで事前学習モデルをファインチューニングしてから，対象とするターゲットタスクで再度ファインチューニングする手法が提案されている[4, 16–18]．これらの手法は STILTs と呼ばれている. ターゲットタスクごとに事前学習モデルを一から作成することは計算コストが非常に高いため，計算コストが低く実装が容易な手法として注目されている．

4 提案手法

4.1 提案手法の概要

日本語 Wikipedia 構造化タスクを，Wikipedia 記事内のパッセージから質問（属性名）の値となる解答（属性値）出力する，解答できない問題を含む機械読解タスクとして解く．本タスクのデータセットでは属性によって属性値が一つも含まれない記事が多数存在し，さらに記事をパッセージに分割するため解答できない問題が非常に多くなる．そこで，解答できない問題を除外し，グループ内の全カテゴリを横断して学習する中間タスクを用いることを提案する．解答できない問題を除外することでデータ件数を削減し，限られた GPU リソース上で実行可能とする．カテゴリを横断して学習することで，訓練データが少ないカテゴリにおいて類似するカテゴリからの学習結果が得られること，解答可能かどうかの判断は行わないことで役割分担ができ，効果的な学習が行われることを狙う．中間タスクでファインチューニングしたモデルを用いて対象のカテゴリのみで学習するターゲットタスクでは，解答できない問題も含めて学習する．

提案手法は Retrospective Reader[15]と同様に役割をわけて学習を行うが，別々のモデルではなく中間タスクとした点と，解答可能かどうかの判断をする 2 値分類タスクではなく機械読解タスクとする点で異なる．予備調査の結果から，2 値分類タスクの場合，正解データに抽出する箇所の情報が含まれないため，学習の際の情報量が不足すると考えられたためである．

BERT の出力は，[13]を参考に，各単語の BIO タグとする．BIO タグは固有表現抽出においてよく用いられる正解ラベルで，B，I，O は，それぞれ単語が解答の先頭，内側，外側にあることを示すものである．

4.2 機械読解用データセットへの変換

Wikipedia 記事内を，HTML タグを分析して小見出しごとに区切ったものをパッセージpとし，pの範囲内に含まれる属性値のデータから属性を質問q，属性値を解答$A = a_1, a_2, \ldots, a_k$とするデータに変換する．$p$に属性値が含まれない属性は$k = 0$となる．$a_i$は$p$内における開始・終了位置情報および，解答の文字列である．表 1 に機械読解用に変換したデータの例を示す．pは HTML タグ中のオプション部分を空白で置き換える処理を行って作成する．

4.3 機械読解モデル

BERT を用いた機械読解モデルは，(p, q, A)を入力として，各単語の BIO タグを出力する．中間タスクとターゲットタスクのモデルは共通である．

BERT への入力系列は，質問qの単語トークン列 $\mathrm{q} = q_1, q_2, \ldots, q_m$およびパッセージ$p$の単語トークン列 $\mathrm{x} = x_1, x_2, \ldots, x_n$を以下のように連結したものとする．

表 2 データ件数

タスク	ターゲットタスク（全カテゴリ合計）			中間タスク (Test のみカテゴリ合計)		
	JP-5	地名	組織名	JP-5	地名	組織名
Train	2,584,533	955,595	1,268,289	272,742	91,909	166,032
Dev	181,889	69,590	90,647	18,026	6,009	10,024
Test	805,238	393,036	519,277	805,238	393,036	519,277

$\{[\mathrm{CLS}], q_1, q_2, \ldots, q_m, [\mathrm{SEP}], x_1, x_2, \ldots, x_n\}$

ここで，[CLS], [SEP] は BERT の特別なトークンである．単語トークン系列$\mathrm{x} = x_1, \ldots, x_j, \ldots, x_n$に対応する BERT の最終層の隠れ状態ベクトル$T_x$を入力とする分類層で，ラベルごとのスコア$S_x = S_{x_1}, \ldots, S_{x_j}, \ldots, S_{x_n}$を出力する．$S_{x_j}$が最も大きいラベルを予測ラベル$y'_{x_j}$とし，正解ラベル $\mathrm{y} = y_{x_1}, \ldots, y_{x_j}, \ldots, y_{x_n}$, $y_{x_j} \in \{\mathrm{B}, \mathrm{I}, \mathrm{O}\}$との交差エントロピー誤差を用いて BERT のファインチューニングを行い，単語トークンごとに 3 値に分類する問題を解く．予測時には予測ラベルy'がB, Iの系列となった単語トークン列を解答候補とし，単語トークン列および単語トークン列のラベルのスコアの平均値をスコアとして出力する．

また，BERT の入力系列には長さの制限があり，最大トークン長以内に収める必要がある．入力系列が最大トークン長よりも長くなる場合，[SEP]の後ろの系列を，設定したトークン数（ストライド）分ずらしながら複数の入力系列を生成する．

4.4 解答候補のルールによる整形

機械読解モデルが出力する解答候補は，以下のルールで整形する．

- 解答文字列前後の HTML タグの削除
- 空文字および漢字以外で 1 文字の解答を除外
- HTML タグ，および記号のみの解答を除外

5 実験

5.1 実験設定

モデルの構築には，公開されているBERTの事前学習済みモデルである「NICT BERT 日本語 Pre-trained モデル BPEあり」[19]（以降，NICT BERT モデル）を利用した．解答可能性付き読解データセット[20]を利用した既存の日本語BERTモデルとの比較[21]において，最も高い性能となったモデルである．システムの実装は，BERTのpytorchの実装[22]を利用した．いくつかのカテゴリにおける事前調査の結果から，バッチサイズは32，学習率は2e-05，トークンの最大長は384，ストライドは128とした．エポック数は最大値を10とし，Devセットにおいて精度が最も良いモデルを選択した．入力文字列はNICT BERTモデルの仕様に合わせ，MeCab-Juman辞書[23]で形態素に分割後，subword-nmt[24]で生成された語彙を用いてsubwordに分割してトークン化した．

5.2 データセット

森羅 2019-JP においてシステム開発用に配布されたアノテーションデータを，提案手法の入力形式に変換した．データセットは訓練（以降 Train），検証(以降 Dev)，テスト(以降 Test)それぞれ，85%，5%，10%の割合で分割し，Train でモデルの訓練，Dev でモデル選択，Test でモデルの評価を行った．生成した BERT 用の入力系列の Train，Dev，Test の件数を表 2 に示す．中間タスクのデータセットは，グループ内のすべてのカテゴリを横断したデータセットである．表 2 で示したように，Train と Dev から解答できない問題を除外することで，おおよそ 10 分の 1 程度のデータ件数になった．

5.3 評価指標

森羅のスコアラー[25]にて算出される属性ごとのF値のマイクロ平均を，各カテゴリの評価指標とする．地名および組織名に関しては，グループに含まれるカテゴリごとのF値のマイクロ平均をさらに平均した値を用いる．

6 結果と考察

6.1 提案手法の有効性

提案手法の有効性を評価するため，森羅2019-JPにおける非公開の100記事を用いた評価型ワークショップでの評価との比較結果[i]を表 3に示す．他のシステムと比較し，提案手法は，JP-5の5カテゴリ，および地名，組織名の平均値において，他の手法を0.01から0.12上回る結果となった．

[i] 全カテゴリの結果が提出されたチームの結果を掲載．

表 3 他の手法との比較結果

		AIP[13]	Nihon_Unisys [11]	NRI-UDI	Tanaka	提案手法
JP-5	空港名	0.856	0.875	0.841	0.825	**0.903**
	市区町村名	0.640	0.633	0.540	0.601	**0.701**
	企業名	0.646	0.595	0.384	0.525	**0.660**
	化合物名	0.475	0.437	0.470	0.466	**0.601**
	人名	0.718	0.755	0.547	0.688	**0.801**
地名		0.574	0.583	0.480	0.529	**0.648**
組織名		0.573	0.504	0.362	0.481	**0.598**

表 4 中間タスクの有効性の評価

モデル		BERT モデル	提案手法
JP-5	空港名	0.819	**0.830**
	市区町村名	0.630	**0.661**
	企業名	0.590	**0.608**
	化合物名	0.518	**0.581**
	人名	0.750	**0.751**
地名		0.578	**0.639**
組織名		0.493	**0.537**

表 5 抽出が難しい属性についての調査結果

カテゴリ／グループ	属性名	BERT モデル	提案手法
化合物名	製造方法	0.113	**0.208**
市区町村名	地名の謂れ	0.102	**0.148**
地名	名前の謂れ（7 カテゴリ平均）	0.098	**0.103**
地名	地名の謂れ（4 カテゴリ平均）	0.038	**0.079**
組織名	名前の謂れ（5 カテゴリ平均）	0.000	**0.049**

6.2 中間タスクの有効性

中間タスクの有効性を評価するため，以下のモデルの比較結果を表 4に示す．

- BERT モデル:中間タスク無しでカテゴリごとにファインチューニングするモデル
- 提案手法 : 中間タスクでファインチューニング後に，カテゴリごとにファインチューニングするモデル

表 4のとおり，提案手法はBERTモデルと比較して，すべてのカテゴリで結果が改善した．訓練データ件数が少ない地名および組織名においても，中間タスクが有効であることが示された．

次に，抽出が難しい属性について抽出能力を向上させることができたかどうかを検証する．抽出が難しい属性として，解答文字列長の平均が100文字以上の属性を対象として調査した結果を表 5に示す．提案手法はBERTモデルと比較して，調査したすべての属性で結果が改善した．地名および組織名の「名前の謂れ」,「地名の謂れ」という属性は，正解データが1カテゴリに平均60件前後と少なく，かつ，解答文字列長は長いことから，特に抽出が難しい属性である．中間タスクにおいて，グループ内の複数のカテゴリの正解データを用いることができたことが精度の改善につながったと考える．また，化合物名の「製造方法」や，市区町村名の「地名の謂れ」については，JP-5グループ内に同じ属性は存在しないにもかかわらず，精度が向上している．中間タスクにおいて解答できない問題を除外したことで，解答可能かどうかの判断はターゲットタスクにまかせ，機械読解能力を効果的に学習できたと考える．

7 おわりに

本論文では，日本語Wikipedia構造化タスクを，解答できない問題を含む機械読解タスクとして定式化し，中間タスクを用いたBERTによる機械読解システムを提案した．中間タスクでは，解答できない問題を除外することでデータ件数を削減し，複数のカテゴリを横断した学習を可能とした．解答が含まれているかどうかの判断をターゲットタスク学習時のみにする施策と，複数のカテゴリを横断することで正解データが少ない属性に対応する施策が有効に働き，これまでの手法よりも優れた性能を達成することが示された．しかしながら，生成した知識データを利用するには，データ件数の少ないカテゴリや属性の抽出精度をさらに改善していく必要がある．類似度の高い属性を集めて学習するなどの工夫を今後の課題としたい．

謝辞

本研究は，森羅プロジェクトからデータを提供していただいて実施しました．森羅プロジェクトの皆様，意見交換させていただいたプロジェクト参加者の皆様に感謝を申し上げます．

参考文献

1. SATOSHI, Sekine, KOBAYASHI, Akio and NAKAYAMA, Kouta. SHINRA: Structuring Wikipedia by Collaborative Contribution. In : *AKBC*. 2019.
2. SEKINE, Satoshi. Extended Named Entity Ontology with Attribute Information. In : *LREC*. 2008.
3. DEVLIN, Jacob, CHANG, Ming-Wei, LEE, Kenton and TOUTANOVA, Kristina. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In : *NAACL*. 2019.
4. PHANG, Jason, FÉVRY, Thibault and BOWMAN, Samuel R. *Sentence Encoders on STILTs: Supplementary Training on Intermediate Labeled-data Tasks*. ArXiv, 2018. ArXiv ID: 1811.01088v2
5. 拡張固有表現階層 定義 Version 8.0.0. http://liat-aip.sakura.ne.jp/ene/ene8/definition_jp/
6. 小林暁雄, 中山功太, 安藤まや and 関根聡. Wikipedia構造化プロジェクト「森羅2019-JP」. In : *言語処理学会第26回年次大会(NLP2020)*. 2020.
7. LI, Xiaoya, FENG, Jingrong, MENG, Yuxian, HAN, Qinghong, WU, Fei and LI, Jiwei. A Unified MRC Framework for Named Entity Recognition. In : *ACL*. 2020.
8. LI, Xiaoya, SUN, Xiaofei, MENG, Yuxian, LIANG, Junjun, WU, Fei and LI, Jiwei. Dice Loss for Data-imbalanced NLP Tasks. In : *ACL*. 2020.
9. LI, Jing, SUN, Aixin, HAN, Jianglei and LI, Chenliang. *A Survey on Deep Learning for Named Entity Recognition*. 2020. arXiv:1812.09449
10. CHEN, Danqi, FISCH, Adam, WESTON, Jason and BORDES, Antoine. Reading Wikipedia to Answer Open-Domain Questions. In : BARZILAY, Regina and KAN, Min-Yen (eds.), *ACL*. 2017.
11. 石井愛. 機械読解によるWikipediaからの情報抽出. In : *言語処理学会第25回年次大会(NLP2019)*. 2019.
12. LIU, Yinhan, OTT, Myle, GOYAL, Naman, DU, Jingfei, JOSHI, Mandar, CHEN, Danqi, LEVY, Omer, LEWIS, Mike, ZETTLEMOYER, Luke and STOYANOV, Veselin. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019. arXiv:1907.11692
13. 中山功太. RoBERTaを用いたQA式属性値抽出システム. In : *森羅2019-JP 最終報告会資料*. 2019.
14. RAJPURKAR, Pranav, JIA, Robin and LIANG, Percy. Know What You Don't Know: Unanswerable Questions for SQuAD. In : *ACL*. 2018.
15. ZHANG, Zhuosheng, YANG, Junjie and ZHAO, Hai. Retrospective Reader for Machine Reading Comprehension. *arXiv*. 2020. arXiv:2001.09694.
16. WANG, Alex, HULA, Jan, XIA, Patrick, PAPPAGARI, Raghavendra, MCCOY, R. Thomas, PATEL, Roma, KIM, Najoung, TENNEY, Ian, HUANG, Yinghui, YU, Katherin, JIN, Shuning, CHEN, Berlin, VAN DURME, Benjamin, GRAVE, Edouard, PAVLICK, Ellie and BOWMAN, Samuel R. Can You Tell Me How to Get Past Sesame Street? Sentence-Level Pretraining Beyond Language Modeling. In : *ACL*. 2019.
17. CLARK, Christopher, LEE, Kenton, CHANG, Ming-Wei, KWIATKOWSKI, Tom, COLLINS, Michael, TOUTANOVA, Kristina and ALLEN, Paul G. BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions. In : *NAACL-HLT*. 2019.
18. SAP, Maarten, RASHKIN, Hannah, CHEN, Derek, LEBRAS, Ronan and CHOI, Yejin. SocialIQA: Commonsense Reasoning about Social Interactions. *EMNLP-IJCNLP*. 22 April 2019.
19. NICT BERT 日本語 Pre-trained モデル. https://alaginrc.nict.go.jp/nict-bert/index.html
20. 鈴木正敏, 松田耕史, 岡崎直観 and 乾健太郎. 読解による解答可能性を付与した質問応答データセットの構築. In : *言語処理学会第24回年次大会(NLP2018)*. 2018.
21. Experiments_on_RCQA. https://alaginrc.nict.go.jp/nict-bert/Experiments_on_RCQA.html
22. WOLF, Thomas, DEBUT, Lysandre, SANH, Victor, CHAUMOND, Julien, DELANGUE, Clement, MOI, Anthony, CISTAC, Pierric, RAULT, Tim, LOUF, Rémi, FUNTOWICZ, Morgan and BREW, Jamie. *HuggingFace's Transformers: State-of-the-art Natural Language Processing*. 2019. arXiv:1910.03771
23. KUDO, T. MeCab : Yet Another Part-of-Speech and Morphological Analyzer. *http://chasen.org/taku/software/mecab/*.
24. GitHub - rsennrich/subword-nmt: Unsupervised Word Segmentation for Neural Machine Translation and Text Generation. https://github.com/rsennrich/subword-nmt
25. GitHub - k141303/shinra_jp_scorer: 森羅2020タスク用のスコアラー. https://github.com/k141303/shinra_jp_scorer