Generative Pre-trained Transformerを利用した会話文完成問題解法

堂坂 浩二 † 高瀬 惇 ‡ 木下 圭 ‡ 石井 雅樹 † 伊東 嗣功 †
† 秋田県立大学 システム科学技術学部 情報工学科
‡ 秋田県立大学 大学院システム科学技術研究科 電子情報システム学専攻
dohsaka@akita-pu.ac.jp

1 はじめに

我々は，言語処理・対話処理技術の高度化を狙いとして，国立情報学研究所が主導する「ロボットは東大に入れるか」プロジェクト [1] の英語問題に取り組み，なかでもセンター入試における会話文完成問題の解法を開発してきた．会話文完成問題は，次の問題例 1 に示すように，会話文中の空所に相応しい文を 4 つの選択肢から選び，会話文を完成させる問題である．この問題では空所 27 に入る正解は選択肢 (1) である．

問題例 1(2015 年度ベネッセ・進研マーク模試 6 月)
Taylor: Are you ready to leave for the baseball game?
Akira: Almost! My guitar lesson ran late and I got home ten minutes ago.
Taylor: Sara's on the phone. She's outside the stadium. What should I tell her?
Akira: 27 We'll never find her in the stadium.
Taylor: I'll also say we'll be there in 20 minutes. Is that OK?
選択肢: (1) Ask her to wait at Gate 11.
(2) I'm at a guitar lesson.
(3) Say we're already at the stadium.
(4) We'll meet her inside.

筆者らが従来提案してきた解法は，主に隣接発話らしさのスコアを使って会話の流れの自然さを算出し，最も自然な会話の流れとなる選択肢を選ぶ方法である [4]．この方法を隣接発話法と呼ぶ．センター入試等の問題を使って評価したところ，隣接発話法の正解率は 45%であった [1, 第 1.4 節].「ロボットは東大に入れるか」プロジェクト [1] の英語において，おおよそ 1 文から構成される短文問題が 90%程度の正解率であることと比べると，複数文から成る会話文完成問題の正解率は低い．理由として，第一に，隣接発話法は空所の直前直後の発話しか見ず，長距離の依存関係の処理が難しいことがある．第二に，隣接発話らしさ認識器の学習データは 35 万発言程度の対話データであり，会話の多様な繋がりを捉えるには十分な量ではない．

一方，自然言語処理の様々なタスクにおいて大規模なデータから事前学習した言語モデルが有効であることが示されている [3, 6]．これらのモデルは，大規模で多様なコーパスから，様々な自然言語処理タスクにとって必要となる長距離の文脈依存関係を捉えるための知識を学習できている可能性がある．そこで，本研究では，OpenAI Generative Pre-trained Transfomer(GPT) [6] の枠組みに沿って，事前学習した Transformer 言語モデルを教師ありの会話文完成問題タスクで fine-tuning するという方法で会話文完成問題の解法を構築した．

以下において，まず，第 2 節で従来の隣接発話法の概要について述べる．第 3 節で GPT による解法について説明し，評価結果を示す．第 4 節で隣接発話法と GPT による解法の比較を行う．

2 隣接発話法

隣接発話法は，会話文完成問題の 4 つの選択肢の各場合について会話の流れの自然さを推定し，最も自然な流れとなる選択肢を選ぶ．会話の流れの自然さは，隣接発話らしさのスコアと感情極性の流れの自然さのスコアの重み付き和として算出した [4]．

隣接発話らしさとは 2 つの発言が会話の中で隣り合って現れる確からしさを表すスコアであり，2 つのスコアのうち主に効果があるのは隣接発話らしさである．隣接発話らしさを求めるため，NTT シチュエーション対話コーパスと Movie-DiC コーパス [2] の 2 種類の対話データからサポートベクトルマシン (SVM) 認識器を学習した．NTT シチュエーション対話コーパスは，会話の場面と話題を指定した上で作業者に対話を作成してもらったものであり，68,020 発言から成る．Movie-DiC コーパスは映画のスクリプトを収集したものであり，本研究では 277,184 発言を使った．特徴量は隣接する会話文の 1,2,3-gram のペアを用いた．

解法の評価は，大学入試センター試験の本試験と追試験，代ゼミセンター模試，ベネッセ模試，独自に収集したその他の問題を合わせた合計 241 問のうち，163

表 1: 従来の隣接発話法の正解率

訓練データ	正解率	
	開発データ	テストデータ
situ	0.37(60/163)	0.40(31/78)
movie	0.36(58/163)	0.40(31/78)
situ-movie-coord	0.39(64/163)	**0.45(35/78)**

問を開発データ，78 問をテストデータに分けたものを使った．開発データで正解率が最大になるように解法のスコアの重みなどのパラメタを調整し，テストデータでの正解率により解法の性能を測った．本稿ではこれらのデータを東ロボ開発・テストデータと呼ぶ．

NTT シチュエーション対話コーパスのみから隣接発話らしさの認識器を学習した場合 (situ), Movie-DiC コーパスのみから認識器を学習した場合 (movie), 各コーパスから学習した 2 つの認識器により別々にスコアを計算し，重み付き和をとる場合 (situ-movie-coord) を比較した [1, 第 1.4 節]．評価結果を表 1 に示す．situ-movie-coord の場合が最も高い正解率 0.45 となった．

3 GPT による解法

3.1 解法の概要

OpenAI GPT [6] の枠組みでは，まず大規模なラベル無しコーパスから Transformer 言語モデルを事前学習する．次にその言語モデルを含意や QA といった教師ありの個別タスクで fine-tuning する．Transformer のモデルとしては decoder のみをもつ多層のモデル [5] が使われている．

Fine-tuning を行う際，個別タスクの問題は，区切り文字を挟んだトークン列に変換され，事前学習された Transformer 言語モデルに渡される．Transformer 言語モデルの最終層の後にタスク固有層を付け加えた構成のネットワークを使って教師あり個別タスクを解き，Transformer 言語モデルとタスク固有層のパラメータを適応させる．

GPT の枠組みに沿って，事前学習された Transformer 言語モデルを教師ありの会話文完成問題タスクで fine-tuning するという方法で解法を実現した．事前学習言語モデルとしては，7000 冊の本からなる BooksCorpus から学習された言語モデル [1]を使った．会話文完成問題は図 1 のように入力系列に変換される．Start, Classify，Delim は区切り文字である．空所を含む発言より前の発言に含まれるトークン列 U_{bef}，空所を含む発言に選択肢を代入した結果生成されるトークン列 Opt_i, 空所を含む発言より後の発言に含まれるトークン列 U_{aft} を区切り文字を挟んで結合することにより，入力系列を生成する．選択肢ごとに生成される 4 つの入力系列を Transformer で独立に処理した後，タスク固有層 (線形変換とソフトマックス層) を経て，解選択のための出力分布を生成する．

[1]https://github.com/openai/finetune-transformer-lm

表 2: GPT による解法の正解率

訓練データ		正解率	
Web/ 問題集	疑似問題	検証データ	テストデータ
無	無	0.38(31/81)	0.38(30/78)
有	無	0.63(51/81)	**0.64(50/78)**
無	有	0.47(38/81)	0.38(30/78)
有	有	0.48(39/81)	0.45(35/78)

3.2 評価実験

GPT による解法を評価するため，以下の環境で評価実験を行った．実装は PyTorch による実装[2]を改良したものを用いた．

- batch サイズ: 8 (1GPU あたりの batch サイズ 4 × GPU 数 2)
- epoch 数: 20

Fine-tuning のための訓練データとして次の会話文完成問題を使った．(A) は常に用い，(B) と (C) の有無で性能を比較した．

(A) 東ロボ開発データの半分の問題: 82 問
(B) ウェブ／問題集から収集した問題: 354 問
(C) NTT シチュエーション対話コーパスから生成した疑似問題: 2,000 問

訓練データ (B) の問題のうち，東ロボ開発・テストデータの各問題との間で，問題文に含まれる単語集合間の Jaccard 係数が 0.4 以上のものを抽出したところ，7 問の問題が抽出された．目視で確認したところ，4 つの問題が東ロボ開発・テストデータの問題と同等のものであったので，訓練データから除いた．

訓練データ (C) を作成するために，まずコーパス中の各対話の最初の 5 ターンのみを切り出した．各対話において一つのターンを空所として扱い，1 つの対話から 5 つの問題を生成した．ターンに元々入っていた発話を正解の選択肢とし，残りの 3 つの不正解の選択肢はコーパス中の他の対話から無作為に抽出した．

検証データとして東ロボ開発データの残り半分 81 問を使い，テストデータとして東ロボテストデータ 78

[2]https://github.com/huggingface/pytorch-openai-transformer-lm

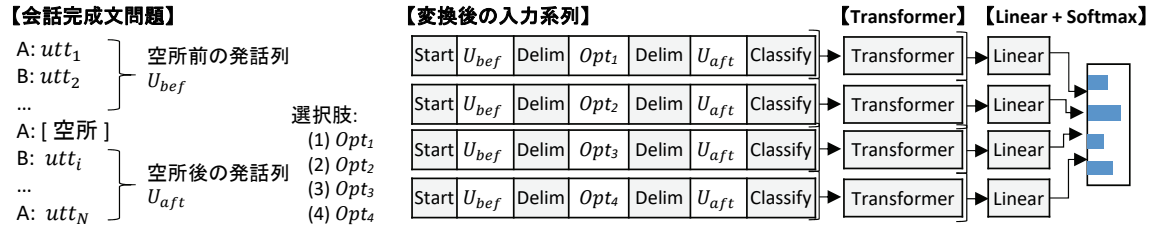

図 1: Fine-tuning のための会話文完成問題の変換

問を使った．検証データの正解率が最大になるときのモデルを選び，テストデータの正解率で評価した．

表 2 に評価結果を示す．ウェブ／問題集から収集した問題 (B) のみを訓練データとして使って fine-tuning したとき，最大の正解率 0.64 となった．大規模データから事前学習した Transformer 言語モデルを少数のタスク固有データで fiine-tuning することで，隣接発話法よりも性能が上がることが分かる．NTT シチュエーション対話コーパスから生成した疑似問題 (C) は有効に働かなかった．今回のような単純な疑似問題生成方式では質の良い負例を作成できないと考えられる．

4 考察

GPT による解法と隣接発話法の其々において，最も正解率の高かった場合を比較して，どのような問題が解けているのかという観点から分析を行った．そのため，テストデータに含まれる問題に関して，回答根拠となる発話の位置と根拠の種類を以下のように分類した．

【回答根拠の発話位置】

1. **直前**: 4 選択肢とも空所直前の発話に根拠がある．
2. **直後**: 4 選択肢とも空所直後の発話に根拠がある．
3. **非隣接**: ある選択肢の根拠が直前・直後以外の発話にあるか，直前と直後の両方に根拠がある．

【回答根拠の種類】

1. **隣接対**: 隣接対 (例:依頼-受諾) の成立の有無が根拠となる．
2. **一貫性**: 隣接対ではない首尾一貫性の有無が根拠となる．
3. **意見**: 対話参加者の意見の一致・不一致の一貫性が根拠となる．
4. **論理**: 発話間の論理的な同値・矛盾が根拠となる．
5. **常識**: 常識，場面の知識を理解した上で一貫性の有無が根拠となる．
6. **多段推論**:共参照や省略を含む会話文を文脈を使って解釈した上で，その解釈内容と別の会話文との間の一貫性の有無が根拠となる．

各問題において根拠となる発話位置，根拠の種類の分類の際には，まず各選択肢に関して分類し，次に問題全体としての分類を行った．また，根拠の種類を分類する際は，各選択肢の根拠の種類のうち，最も大きな番号をもつものを問題の根拠の種類とした．

分類について問題例を使って説明する．次の問題例 2 は根拠となる発話位置を直前，根拠の種類を隣接対と分類した．正解は選択肢 (4) である．空所直前の “I’m sorry” と正解選択肢 (4) の “Don’t worry about that” は隣接対と言える．不正解選択肢 (1) の “don’t mind” は “I’m sorry” と隣接対として呼応しない．他の選択肢も同様である．

問題例 2(1995 年度センター入試追試)
A: I’ve come to see you because I want to apologize.
B: I can’t imagine what for.
A: Well, to say I’m sorry for losing my temper at the meeting.
B: 28
選択肢: (1) All right, don’t mind.
(2) Next time, put it where you can find it.
(3) Not at all, you’re very welcome.
(4) Oh, I see. Don’t worry about that.

これに対して，第 1 節で示した問題例 1 は，根拠となる発話位置を非隣接，根拠の種類を一貫性と分類した．不正解選択肢 (2), (4) を排除する根拠は，空所直前の発話 “What should I tell her?” と隣接対として呼応しないことと考えられるが，不正解選択肢 (3) を排除するためには対話全体の状況と一貫しないことを認識する必要がある．

次の問題例 3 は，根拠となる発話位置を非隣接，根拠の種類を常識と分類した．正解の選択肢は (2) である．テレビがうるさくて聞こえないから聞き返すという場面の知識があって解ける問題である．

問題例 3(1993 年度センター入試追試)
A: Have you cleaned your room yet?
B: Sorry, the TV is too loud. 32
A: Did you clean up your room?
選択肢: (1) No, I haven’t.
(2) Now, what did you say?
(3) Now, what do you mean?
(4) Yes, an hour ago.

表 3: 回答根拠の発話位置の比較

発話位置	GPT による解法	隣接発話法
直前	0.80 (4/5)	0.80 (4/5)
直後	0.92 (11/12)	0.58 (7/12)
非隣接	0.57 (35/61)	0.39 (24/61)

次の問題例 4 は，根拠となる発話位置を非隣接，根拠の種類を多段推論と分類した．正解の選択肢は (4) である．第 2 文の前半が第 1 文を参照しており，その内容を文脈を使って解釈した上で，第 2 文の前半と後半が矛盾することを認識する必要がある．なお，根拠の種類のうち意見・論理については例を省略する．

問題例 4(1999 年度センター入試)
A: What are your plans for this weekend?
B: I haven't really thought about it. 7
A: I'm thinking of going to the beach. Want to come?
選択肢: (1) How do you plan to go?
(2) I'm planning to go mountain climbing.
(3) I've booked our room.
(4) Why do you ask?

表 3 と表 4 に，GPT による解法 (正解率 0.64) と隣接発話法 (正解率 0.45) に関して，根拠となる発話位置，根拠の種類ごとの正解率を示す．まず，根拠の発話位置に関して，根拠の発話位置が直前の場合は両解法ともほぼ解けている．根拠の発話位置が直後の場合，GPT による解法の方がより確実に問題を解けており，根拠の発話位置が非隣接の場合，GPT による解法が隣接発話法に優っていることが分かる．

次に，根拠の種類に関して，隣接対の有無が根拠になる場合は，そもそも例は少ないが，隣接発話法も GPT による解法も確実に解けていると言える．一貫性の有無が根拠になる場合は，隣接発話法の正解率が 45%であるのに対して，GPT による解法の正解率は 88%である．大規模データにより事前学習した言語モデルを会話文完成問題タスクで fine-tuning することにより，長距離の依存を理解するための知識の学習が行われたと言える．

しかし，意見・論理・常識・多段推論といった，意見の一致・不一致の理解，論理的な矛盾の発見，言語外の常識の利用，多段階の言語処理等が必要な問題には，GPT による解法も対処が難しいことが示唆される．

5 おわりに

本研究では，OpenAI GPT[6] の枠組みに沿って，事前学習された Transformer 言語モデルを教師ありの会話文完成問題タスクで fine-tuning するという方法で，会話文完成問題の解法を開発した．空所に隣接した発話のみを考慮する従来の隣接発話法と比較して，GPT による解法は大幅に性能が向上し，長距離の依存を理解するための知識の学習が行われたと言える．しかし，分析の結果，言語外の常識や多段の言語タスクの適用が必要な問題には GPT による解法は対処が難しいことも分かった．今後は，他の言語処理手法と組み合わせることによって，解法の改良を図る．

表 4: 回答根拠の種類の比較

根拠の種類	GPT による解法	隣接発話法
隣接対	1.00 (4/4)	1.00 (4/4)
一貫性	0.88 (45/51)	0.45 (23/51)
意見	0.00 (0/5)	0.40 (2/5)
論理	0.33 (1/3)	0.67 (2/3)
常識	0.00 (0/11)	0.27 (3/11)
多段推論	0.00 (0/4)	0.25 (1/4)

謝辞

本研究を推進するにあたって，大学入試センター試験問題のデータをご提供下さった独立行政法人大学入試センターおよび株式会社ジェイシー教育研究所，ならびに実験データをご提供くださいました学校法人高宮学園，株式会社ベネッセコーポレーションに感謝いたします．また，NTT コミュニケーション科学基礎研究所には NTT シチュエーション対話コーパスをご提供いただきました．謹んで感謝の意を表します．

参考文献

[1] 新井紀子, 東中竜一郎（編）. 人工知能プロジェクト「ロボットは東大に入れるか」: 第三次 AI ブームの到達点と限界. 東京大学出版会, 2018.

[2] Rafael E. Banchs. Movie-DiC: A movie dialogue corpus for research and development. In *Procs. of ACL*, pp. 203–207, 2012.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[4] 堂坂浩二, 坂本祐磨, 高瀬惇. 隣接発話らしさを利用した英語会話文完成問題の回答手法. 2016 年度人工知能学会全国大会, 1K3-4, 2016.

[5] Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating wikipedia by summarizing long sequences. In *Proc. ICLR*, 2018.

[6] Alec Radford, Karthik Narasimhan, Time Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. Technical report, OpenAI, 2018.