語りに傾聴を示す応答タイミングの検出のためのテキストデータの利用

渡邉優 [1]　伊藤滉一朗 [2]　松原茂樹 [2,3]
[1] 名古屋大学情報学部　[2] 名古屋大学大学院情報学研究科
[3] 名古屋大学情報基盤センター
watanabe.yu.x3@s.mail.nagoya-u.ac.jp　ito.koichiro.v1@s.mail.nagoya-u.ac.jp
matsubara.shigeki.z8@f.mail.nagoya-u.ac.jp

概要

会話エージェントが人間に代わって語りの聴き手を担うことが期待されている．これらが聴き手として認められるには，傾聴を示す目的で語りに応答する発話である傾聴応答を適切なタイミングで生成することが効果的である．本論文では，傾聴応答タイミングの検出のためのテキストデータの利用について述べる．事前学習済み言語モデルを傾聴応答タイミング検出タスクで fine-tuning する前に，テキストへの句読点挿入タスクを中間タスクとして導入する．応答タイミングの検出実験の結果，中間タスクの導入により，特に，傾聴応答のデータ量が十分ではない場合に，モデルが傾聴応答タイミングの特徴を効率的に学習できることを確認した．

1　はじめに

日本では，独居高齢者の増加など，社会の個人化が進行し [1]，人が語れる機会が失われつつある．語ることは人間の基本的な欲求であり，語る機会の消失は社会問題といえる．この問題の解決策の 1 つは，コミュニケーションロボットやスマートスピーカーなどの会話エージェントが，人間に代わって語りの聴き手を担い，話し手に語る機会を提供することである．これらが語りの聴き手として認められるには，「相手の語りに耳を傾けて聴く」という傾聴の態度を示すことが重要である．そのための明示的な手段は，相手の語りに応答することである．以降では，傾聴を示す目的で語りに応答する発話を**傾聴応答**と呼ぶ．

傾聴応答は，語りを聴いていることを伝えるとともに，相手への理解を示す働きを持つ．適切なタイミングで生成できれば，話し手の語る意欲を高められるものの，不適切なタイミングでの生成は語りを遮ることになり，逆効果になりうる [2]．したがって，語りの聴き手を担う会話エージェントの実現のためには，傾聴応答タイミングを適切に検出できる必要がある．語りと傾聴応答のデータに基づく検出手法の開発が現実的であるが，そのデータの蓄積は必ずしも十分ではなく，その収集も容易ではない．

語り	傾聴応答
20 代の後半に 仲良し 3 人組で 安い宿を使って 軽井沢に 行こうと	ええ 軽井沢に

図 1　語りと傾聴応答の例

そこで本論文では，傾聴応答タイミングの検出のためのテキストデータの利用の効果を考察する．これまでに，傾聴応答タイミング検出におけるテキストデータ利用の効果が報告されているものの [3]，傾聴応答のデータ量がテキストデータ利用の効果に及ぼす影響は明らかにされていない．そこで本研究では，テキストへの句読点挿入タスクを中間タスク [4, 5] として導入し，傾聴応答のデータ量と中間タスク導入の効果との関係を考察する．傾聴応答タイミングの検出実験を実施し，中間タスクの導入によって，特に，傾聴応答のデータ量が十分ではない場合に，傾聴応答タイミングの特徴を効率的に学習できることを確認した．

2　傾聴応答

傾聴応答は，語りを聴いていることを伝えるとともに，相手への理解を示す働きを持つ．図 1 に語りと傾聴応答の例を示す．傾聴応答を適切なタイミングで生成できれば，語り手の語る意欲を高められる

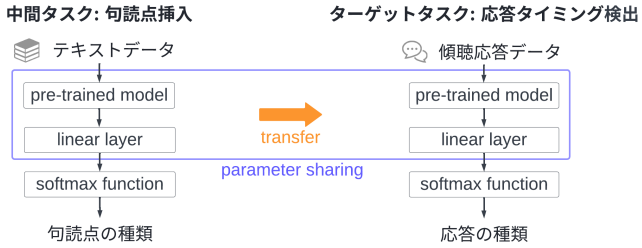

図 2　テキストデータの利用の概略

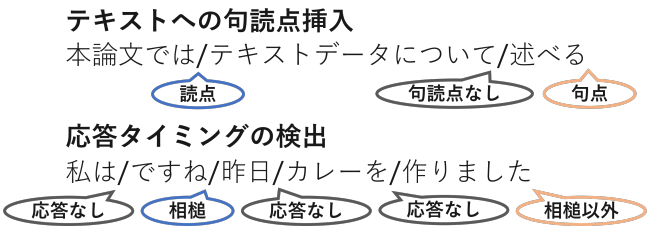

図 3　各タスクの例

ものの，不適切なタイミングでの生成は語りを遮ることになり，逆効果になりうる [2].

これまで，傾聴応答の代表例である相槌のタイミング検出手法として，ルールベースによる手法 [6]，n-gram モデルによる手法 [7]，決定木による手法 [8]，CRF による手法 [9]，SVM による手法 [10]，LSTM による手法 [11] などが提案されている．近年では，事前学習済みモデルによる手法 [12] も提案されている．

3　テキストの句読点

句読点は，テキストの区切りを明示する記号であり，文の可読性の向上や読み手による文の理解に影響を与える [13]．句読点を適切な位置に挿入できれば，読み手によるテキストの理解が容易になるものの，不適切な位置への挿入は，テキストを読み進めることの妨げとなり，逆効果になりうる．

これまでに，テキストへの句読点挿入手法として，n-gram 言語モデルによる手法 [14]，CRF による手法 [15]，最大エントロピー法による手法 [13]，RNN による手法 [16] などが提案されている．近年では，事前学習済みモデルによる手法 [17] も提案されている．

4　応答タイミング検出のための句読点の利用

事前学習済みモデルの fine-tuning に関して，ターゲットタスクで fine-tuning する前に中間タスクで fine-tuning することで，ターゲットタスクに有用な特徴を効果的に学習させる手法が存在する [4, 5]．本研究では，応答タイミング検出の中間タスクとして，テキストへの句読点挿入タスクを導入する．

図 2 と図 3 上に，本論文におけるテキストデータの利用の概略と，テキストへの句読点挿入の例をそれぞれ示す．図 3 において “/” は文節境界を示す．句読点挿入タスクでは，テキストにおける各文節境界を，読点の挿入位置，句点の挿入位置，句読点なしの位置の 3 クラスへ分類する．分類のための入力には，分類対象の文節境界以前の文字列を用いる．

ターゲットタスクでは，語りの文節境界を傾聴応答タイミングの候補とし，各文節境界が傾聴応答タイミングであるか否かを判定する．傾聴応答には，相槌を始めとして，感心，繰り返しなど，いくつか種類が存在している [18]．本研究では，傾聴応答の代表例である相槌と，相槌以外の傾聴応答に分けて，そのタイミング検出を行う．すなわち，各文節境界を，相槌のタイミング，相槌以外のタイミング，応答なしのタイミング（傾聴応答ではないタイミング）の 3 クラスへ分類する．分類のための入力には，分類対象の文節境界以前の語りの文字列を用いる．図 3 下に，応答タイミングの検出の例を示す．本研究では，相槌と読点の使用頻度に着目し，相槌のタイミングと読点の挿入位置，相槌以外のタイミングと句点の挿入位置，応答なしのタイミングと句読点なしの位置が，それぞれ対応しているものとみなす．

5　応答タイミングの検出実験

5.1　実験概要

これまでに，傾聴応答タイミング検出におけるテキストデータ利用の効果が報告されているものの [3]，傾聴応答のデータ量に関するテキストデータ利用の効果は明らかにされていない．そこで本実験では，ターゲットタスクである傾聴応答タイミング検出のための学習データ量ごとに，中間タスク導入の効果を評価する．ターゲットタスクの学習データ量の設定は下記の通りである．

- **実験設定 (a)**：学習データ量を 200 から 2,000 まで，200 ずつ増加させる．
- **実験設定 (b)**：学習データ量を 2,000 から 16,000 まで，2,000 ずつ増加させる．

ターゲットタスクのみで fine-tuning されたモデルを baseline，両方のタスクで fine-tuning されたモデルを

表 1　京都大学テキストコーパスと傾聴応答コーパスにおける各クラスの出現分布

	京都大学テキストコーパス			傾聴応答コーパス		
	読点	句点	句読点なし	相槌	相槌以外	応答なし
学習データ	22,786	16,731	123,846	5,227	3,324	8,866
開発データ	1,648	1,172	8,540	1,703	1,215	3,352
テストデータ	2,444	1,836	13,014	1,657	1,152	3,350

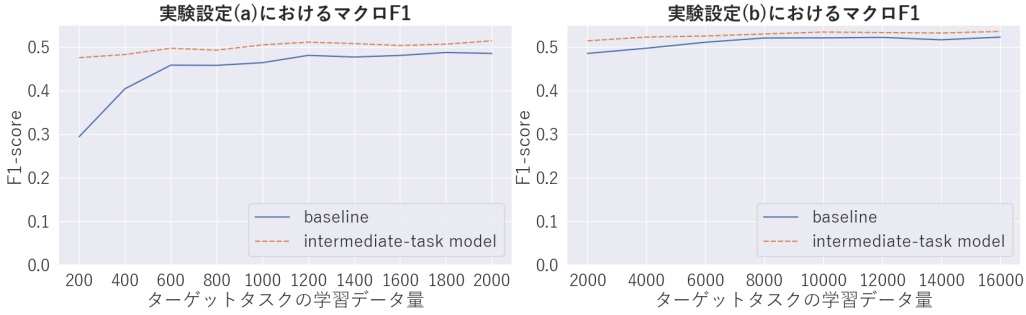

図 4　実験設定 (a) と (b) におけるマクロ F1

intermediate-task model と表記する．学習と評価は乱数のシードを変えて 3 回行い，そのマクロ F1 の平均値を最終的な評価値とした．

5.2　実験データ

中間タスク用のテキストデータとして，京都大学テキストコーパス [19] を用いる．このテキストコーパスを，学習，開発，テスト用に分割した．表 1 にデータの規模を示す．

ターゲットタスク用のデータとして，傾聴応答コーパス [20] を用いる．傾聴応答コーパスは，高齢者のナラティブコーパス JELiCo[21] に，複数の聴き手の傾聴応答が独立に付与されたデータである．本実験では，聴き手 1 名分の傾聴応答を用いた．

傾聴応答コーパスに含まれる語りを，CaboCha[22] によって文節に分割した．さらに各応答を，その発声開始時刻と最も近い位置にある文節境界に対応付けた．相槌のみが対応付いた文節境界を相槌タイミング，1 つ以上の相槌ではない応答が対応付いた文節境界を相槌以外のタイミング，1 つも応答が対応付かなかった文節境界を応答なしのタイミングとする．傾聴応答コーパスを学習，開発，テスト用に分割した．表 1 にデータの規模を示す．

5.3　実装

本実験では，事前学習済みの BERT[1] に，3 クラス用の分類層を追加することで，句読点挿入及び傾聴応答タイミング検出を行う．ターゲットタスクでの学習では，中間タスクで学習されたパラメータを初期値として，分類層を含む全パラメータの更新を続ける．BERT への入力は，先頭の [CLS] トークンに，分類対象の文節境界の直前 3 つの文節の文字列を連接したものとした．損失関数には Cross Entropy Loss を，最適化手法には AdamW を用いた．中間タスクの評価データでのマクロ F1 が最良となったエポックのモデルを，ターゲットタスクでさらに学習する．ターゲットタスクに対する性能評価には，開発データでのマクロ F1 が最良となったエポックのモデルを用いた．ハイパーパラメータと実装に用いたライブラリの詳細は，付録 A を参照されたい．

5.4　実験結果

ターゲットタスクの結果について報告する [2]．図 4 に，実験設定 (a) と (b) の結果をそれぞれ示す．いずれの設定においても，学習データ量によらず，intermediate-task model が baseline を上回った[3]．このことから，テキストデータを用いた句読点挿入タスクでの fine-tuning によって，傾聴応答タイミングの検出性能が向上することを確認した．また，学習データが少量である設定 (a) において，intermediate-task model が baseline を大きく上回る傾向にあった．したがって，傾聴応答タイミング検出タスクのデータが少量である場合には特に，テキストデータを用いた句読点挿入タスクでの fine-tuning が有効であるといえる．

1)　https://huggingface.co/cl-tohoku/bert-base-japanese-v3

2)　中間タスクの結果は付録 B を参照されたい．
3)　有意差検定の結果は付録 C を参照されたい．

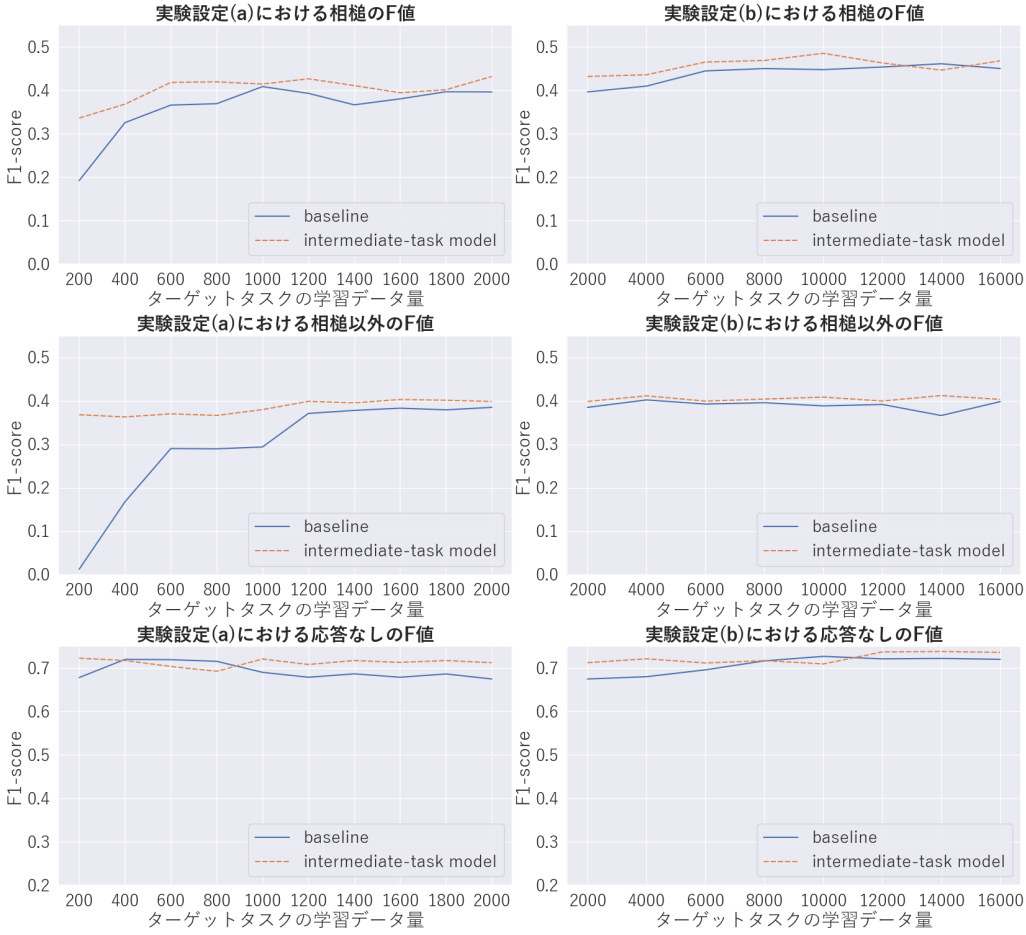

図 5　実験設定 (a) と (b) における各クラスの F 値

5.5　考察

ターゲットタスクの 3 クラスについて，中間タスク導入の効果を個別に考察する．まず，相槌における中間タスク導入の効果について述べる．図 5 の上段に，実験設定 (a) と (b) の結果をそれぞれ示す．学習データ量が 12,000 程度までは，intermediate-task model が baseline を上回り，それ以降では，baseline を下回ることもあった．このことから，ターゲットタスクの学習データ量が十分である場合には，中間タスク導入の効果は限定的であるものの，学習データ量が十分とはいえない場合には，中間タスク学習が効果的であるといえる．

次に，相槌以外における中間タスク導入の効果について述べる．図 5 の中段に，実験設定 (a) と (b) の結果をそれぞれ示す．学習データ量が 1,000 程度までは，intermediate-task model が baseline を大きく上回った．学習データ量が 1,000 を超えると差は小さくなるものの，性能向上を確認できた．このことから，相槌と相槌以外では，中間タスクの導入が有効な状況や，その効果の大きさが異なるといえる．

最後に，応答なしにおける中間タスク導入の効果について述べる．図 5 の下段に，実験設定 (a) と (b) の結果をそれぞれ示す．相槌と相槌以外の結果とは異なり，学習データ量によらず，intermediate-task model と baseline に大きな差はなかった．このことから，中間タスクの導入は，応答なしの性能を維持したまま，相槌と相槌以外の性能向上を達成しているといえる．

6　まとめ

本論文では，傾聴応答タイミングの検出のためのテキストデータの利用について述べた．テキストへの句読点挿入タスクを中間タスクとして導入し，その効果を考察した．実験の結果，中間タスクの導入によって，特に，傾聴応答のデータ量が十分ではない場合に，傾聴応答タイミングの特徴を効率的に学習できることを確認した．今後は，傾聴応答の種類と読点の用法の種類の関係性を考察し，より効率的な学習方法を検討したい．

謝辞

高齢者のナラティブコーパスは，奈良先端科学技術大学院大学ソーシャル・コンピューティング研究室から提供いただいた．本研究は，一部，名古屋大学のスーパーコンピュータ「不老」の一般利用制度により実施した．

参考文献

[1] Ministry of Health, Labor and Welfare. Summary report of comprehensive survey of living conditions 2019, 2020. https://www.mhlw.go.jp/english/database/db-hss/dl/report_gaikyo_2019.pdf.

[2] 大谷佳子. 対人援助の現場で使える 聴く・伝える・共感する技術 便利帖. 翔泳社, 2017.

[3] 長連成, 越智景子, 井上昂治, 河原達也. 大規模テキストデータを用いた事前学習による音声対話の相槌予測. 情報処理学会第 84 回全国大会講演論文集, pp. 305–306, 2022.

[4] Jason Phang, Thibault Févry, and Samuel R. Bowman. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. **arXiv preprint arXive:1811.01088**, 2019.

[5] Ting-Yun Chang and Chi-Jen Lu. Rethinking why intermediate-task fine-tuning works. In **Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2021**, pp. 706–713, 2021.

[6] Nigel Ward and Wataru Tsukahara. Prosodic features which cue back-channel responses in english and japanese. **Journal of Pragmatics**, Vol. 32, No. 8, pp. 1177–1207, 2000.

[7] Nicola Cathcart, Jean Carletta, and Ewan Klein. A shallow model of backchannel continuers in spoken dialogue. In **Proceedings of the 10th Conference on European Chapter of the Association for Computational Linguistics**, Vol. 1, pp. 51–58, 2003.

[8] Norihide Kitaoka, Masashi Takeuchi, Ryota Nishimura, and Seiji Nakagawa. Response timing detection using prosodic and linguistic information for human-friendly spoken dialog systems. **Transactions of the Japanese Society for Artificial Intelligence**, Vol. 20, No. 3, pp. 220–228, 2005.

[9] Louis-Philippe Morency, Iwan de Kok, and Jonathan Gratch. A probabilistic multimodal approach for predicting listener backchannels. **Journal of Autonomous Agents and Multi-Agent Systems**, Vol. 20, No. 1, pp. 70–84, 2010.

[10] 大野誠寛, 神谷優貴, 松原茂樹. 対話コーパスを用いた相づち生成タイミングの検出. 電子情報通信学会論文誌, Vol. J100-A, No. 1, pp. 53–65, 2017.

[11] Robin Ruede, Markus Müller, Sebastian Stüker, and Alex Waibel. Enhancing backchannel prediction using word embeddings. In **Proceedings of the 18th Annual Conference of the International Speech Communication Association**, pp. 879–883, 2017.

[12] Jin Yea Jang, San Kim, Minyoung Jung, Saim Shin, and Gahgene Gweon. BPM_MT: Enhanced backchannel prediction model using multi-task learning. In **Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing**, pp. 3447–3452, 2021.

[13] 村田匡輝, 大野誠寛, 松原茂樹. 読点の用法的分類に基づく日本語テキストへの自動読点挿入. 電子情報通信学会論文誌, Vol. J95-D, No. 9, pp. 1783–1793, 2012.

[14] Agustin Gravano, Martin Jansche, and Michiel Bacchiani. Restoring punctuation and capitalization in transcribed speech. In **Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing**, pp. 4741–4744, 2009.

[15] Wei Lu and Hwee Tou Ng. Better punctuation prediction with dynamic conditional random fields. In **Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing**, pp. 177–186, 2010.

[16] Ottokar Tilk and Tanel Alumäe. Bidirectional recurrent neural network with attention mechanism for punctuation restoration. In **Proceedings of the 17th Annual Conference of the International Speech Communication Association**, pp. 3047–3051, 2016.

[17] Micha l Pogoda and Tomasz Walkowiak. Comprehensive punctuation restoration for English and Polish. In **Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2021**, pp. 4610–4619, 2021.

[18] 日本語記述文法研究会. 現代日本語文法 7. くろしお出版, 2009.

[19] Sadao Kurohashi and Makoto Nagao. Building a japanese parsed corpus while improving the parsing system. In **Proceedings of the 1st International Conference on Language Resources and Evaluation**, pp. 719–724, 1998.

[20] Koichiro Ito, Masaki Murata, Tomohiro Ohno, and Shigeki Matsubara. Construction of responsive utterance corpus for attentive listening response production. In **Proceedings of the 13th Language Resources and Evaluation Conference**, pp. 7244–7252, 2022.

[21] Kenji Aramaki. Japanese elder's language index corpus v2, 2017. https://figshare.com/articles/dataset/Japanese_Elder_s_Language_Index_Corpus_v2/2082706/1.

[22] Taku Kudo and Yuji Matsumoto. Japanese dependency analysis using cascaded chunking. In **Proceedings of the 6th Conference on Natural Language Learning**, pp. 63–69, 2002.

A　モデルのハイパーパラメータと実装に用いたライブラリ

モデルの実装には，pytorch [4] と huggingface の Trainer [5] を用いた．中間タスクでは 4 枚の GPU を用いて分散学習を行い，ターゲットタスクでは 1 枚の GPU で学習を行った．表 2 に，学習の詳細設定を示す．これらのハイパーパラメータの値は，各タスクにおける開発データを用いて定めた．

表 2　学習設定

	中間タスク	ターゲットタスク
num_train_epochs	30	20
learning_rate	1e-5	5e-6
per_device_train_batch_size	32	16
gradient_accumulation_steps	4	1
earlystopping	True	True
patience	5	5

B　中間タスクの結果

中間タスクである句読点挿入タスクの結果について報告する．表 3 に，テストデータに対する句読点の挿入性能として，各クラスの適合率，再現率，F 値を示す．いずれのクラスにおいても，高い挿入性能を示していることを確認した．また，マクロ F1 は 0.800 であった．したがって，モデルは中間タスクを通して，テキストにおける句読点の挿入位置の特徴を適切に学習できたものと考えられる．

表 3　中間タスクである句読点挿入タスクの結果

	適合率	再現率	F 値
読点	0.754	0.640	0.692
句点	0.757	0.801	0.779
句読点なし	0.920	0.938	0.929

C　有意差検定

5.4 節の実験結果における有意差について報告する．ボンフェローニ補正を適用して，McNemar 検定を多クラスに拡張した McNemar-Bowker 検定 ($\alpha = 0.05$) を行った．全ての学習データ量において 3 回有意差が認められた．

4）https://pytorch.org/
5）https://huggingface.co/docs/transformers/main_classes/trainer