話者情報を認識した対話要約

楢木悠士
早稲田大学基幹理工学部情報理工学科
yuji.1277@akane.waseda.jp

酒井哲也
早稲田大学基幹理工学部情報理工学科
tetsuyasakai@acm.org

1 導入

自動対話要約は対話文書から話者の意図を客観的に捉え，正確かつ簡潔な要約を生成することを目的としたタスクである．実世界におけるアプリケーションへの応用などの需要が増加している．例えば，顧客サービスセンターや病院ではそのやりとりを要約できれば有用であると考えられる．しかし，対話要約のための高品質なデータセットは公開されていなかったため，対話要約の研究は未だ少ない． 特に深層学習のための適切で十分に大きい公開データセットは SAMSum[1] の登場までなかったため，対話に特化された深層学習の研究は非常に少ない．本研究では対話特有の話者情報に焦点を当て，Transformer[2] を基にしたモデルのための対話に特化したアーキテクチャを提案する．はじめに，話者交代情報を含んだ Turn Embedding と話者識別情報を含んだ Speaker Embedding を提案する．そして，それらの Embedding を一部の次元に限定する Partial Turn Embedding と Partial Speaker Embedding を提案する．実験の結果，Partial Turn/Speaker Embedding は文書要約における最新のベースラインと比較して，収束性と生成要約の ROUGE スコアの向上を示した．

2 関連研究

文書要約の代表的な手法をいくつか紹介する．Pointer-Generator[3] は RNN をベースにした Sequence-to-Sequence(Seq2Seq) モデルである．Pointer-Generator は以前から課題であった辞書に含まれない単語に対応できない Out-of-Vocabulary 問題や，RNN による自然言語生成で頻繁に引き起こされる文字列の無意味な反復を解決した．Pointer-Generator 以降は Transformer を基にした手法が登場してきた．文書要約は長文の意味を捉える必要があるタスクであり，入力シーケンスの遠い位置の関係を捉えることができる Transformer は要約に適していると考えられる．Liu らは事前学習を行う BERT を文章要約に応用した BERTSum, BERTSumAbs を提案した [4]．BERTSum は事前学習可能なエンコーダーを用いて抽出型要約を可能にした．BERTSumAbs は BERTSum のエンコーダーと Transformer を基にしたデコーダーを用いることで，抽象型要約を可能にした．ただし，BERTSum で事前学習できるパラメータはエンコーダーに限られる．Zhang らが提案した PEGASUS は Gap Sentences Generation(GSG) と呼ばれる事前学習手法を用いた抽象型要約手法である [5]．通常の BERT の事前学習に用いられる Masked Language Model では，入力文章の一部のトークンをマスクし，そのマスク箇所を予測する手法である．一方，GSG では入力文章の一部の文をマスクし，マスクされた文全体を予測する．この手法により，デコーダーも事前学習ができ，文生成に特化した学習が可能となった．

次に，対話要約のデータセットとそれらを用いた研究をいくつか紹介する．Goo らは RNN を用いて対話における発言の役割を副次的に学習する抽象型要約モデルを提案した [6]．Goo らの使用した AMI 会議コーパス [7] は発言に役割がアノテーションされたデータセットであり，要約文が短いことが特徴的である．Yuan らは対話のドメインを副次的に学習する Pointer-Generator を基にした抽象型要約モデル SPNet を提案した [8]．ここで用いた MultiWOZ-2.0[9] は対話のドメインがアノテーションされた対話コーパスであり，対話要約のためのデータセットではない．そこで Yuan らはクラウドソーシングにより構築した要約を用いた．最後に，Gliwa らは深層学習に十分な大きさの高品質な対話要約データセットが公開されていないという課題を解決するために SAMSum を公開した [1]．SAMSum は言語学者によって人手で構築されたデータセットであり，世間話や会議の手配などあらゆる日常的な会話で構成されている．

BERTSum や PEGASUS などの Transformer を基にしたモデルの入力には，入力文から生成されるい

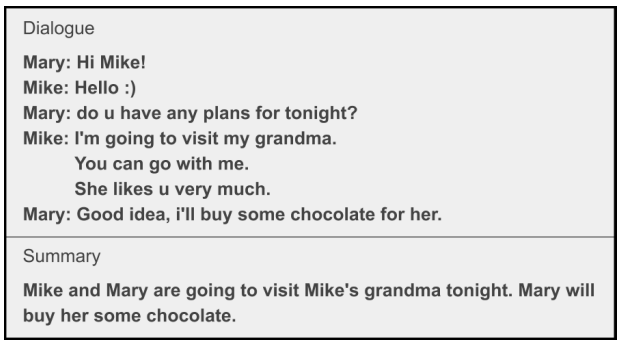

図 1　SAMSum の対話文と要約の例

くつかの種類の Embedding の和を用いる．一般的に BERT に用いられる Embedding には 3 種類ある．入力文の各トークンを表す Token Embedding，入力文の 2 種類の区分を表す Segment Embedding，入力文の位置を表す Position Embedding がある．Token Embedding と Position Embedding は自然言語処理の主なタスクにおいて共通する手法で生成される．しかし，Segment Embedding の生成に用いられる入力文の 2 つの区分はタスクによって様々である．質問応答タスクにおいては質問とパラグラフで分けられ，BERTSum では奇数番目の文と偶数番目の文で Embedding を分けている．タスクによっては区分がなく，Segment Embedding を用いない場合も多い．チャットボットにおける返答選定タスクにおいては，対話と返答で異なる Segment Embedding を用いる．以上 3 種類の Embedding に加えて，Gu らは Speaker Embedding を追加で加算することで，返答選定タスクの性能の向上をもたらした [10]．Gu らの提案した Speaker Embedding は Segment Embedding と同様の構造を持ち，話者交代ごとに 2 つの区分を入れ替える Embedding である．

3　話者情報を与える Embedding

3.1　データ前処理

本研究では，モデルの学習に十分な量の高品質なデータを持つ SAMSum を用いて学習を行う．SAMSum の対話文と要約の例を図 1 に示す．本研究では，対話特有の特徴を捉えるため，Gu らの用いた特殊トークンを参考に 3 つの特殊トークン挿入した．特殊トークンの名称と挿入箇所の対応を表 1 に示す．さらに，それら 3 種類の特殊トークンを挿入した入力例を図 2 に示す．

表 1　特殊トークン名と挿入箇所

トークン名	挿入場所
[SAYS]	話者名と発言の間
[EOU]	すべての発言の後 (End-of-Utterance)
[EOT]	毎回の最後の発言の後 (End-of-Turn)

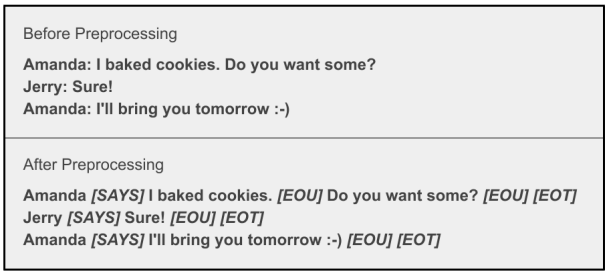

図 2　前処理前と前処理後の対話例

3.2　Turn/Speaker Embedding

本研究では対話要約モデルとして最先端の文書要約モデルである PEGASUS を用いる．PEGASUS は Transformer を基にしたモデルであり，入力には対話文から得られる Token Embedding と Position Embedding を用いる．それらに加えて，Gu らの BERT への入力に話者情報を Embedding に加えるというアイデアを基に，我々は話者情報を含んだ追加の Embedding を 2 種類提案する．話者交代情報を示す Turn Embedding と話者識別情報を示す Speaker Embedding である．ただし，Gu らの提案した Speaker Embedding と我々が提案する Speaker Embedding は異なる．

Turn Embedding は Gu らの提案した Speaker Embedding と同様の方法で構成されるもので，2 つの Embedding を持ち，話者が入れ替わるごとに Embedding を入れ替える．それに対して，我々の提案する Speaker Embedding は，話者ごとに Embedding を変更する．Speaker Embedding が持つ Embedding の種類は SAMSum に含まれる対話の最大話者数に対して十分に大きい 15 種類とした．構造上の欠点として，Embedding が十分に学習されるかどうかは学習データに含まれる話者数の分布に依存する．学習データに n 人の対話が含まれなければ，n 番目の Embedding は学習されないことになる．

本研究では話者交代情報または話者識別情報を包含した追加の Embedding を提案し，PEGASUS に適応させた時の影響を実験的に分析した．したがって，モデルの入力に用いる Embedding は

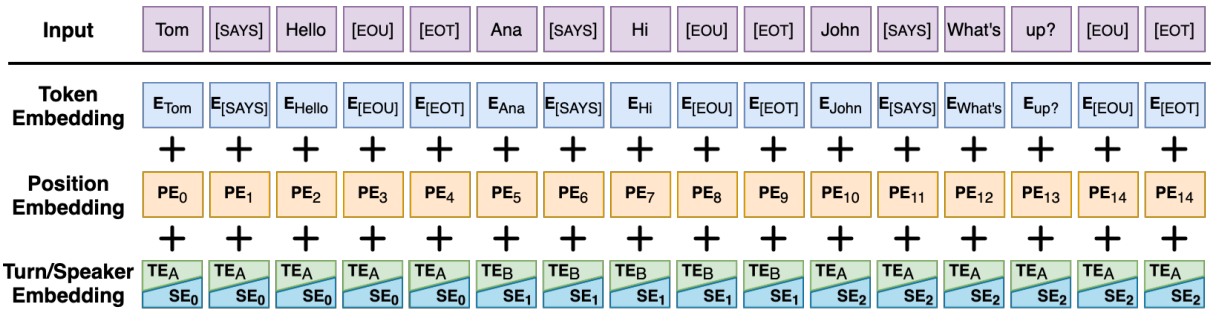

図 3　入力表現のアーキテクチャ

Token Embedding と Position Embedding に加え，Turn Embedding または Speaker Embedding の 3 つの和である．入力表現のアーキテクチャを図 3 に示す．

3.3　Position Embedding の次元方向の情報量

本研究では，モデルに入力する Embedding に追加の Embedding を用いることで話者情報をモデルに認識させる．我々は Embedding の効果を最大化するため，Embedding の次元方向の工夫を行った．

実験に用いる 3 種類の Embedding のうち，入力シーケンスの位置情報を表す Position Embedding には Sinusoidal Positional Embedding[2] を用いる．これは固定のパラメータを持つ Embedding であり，式 1, 2 によって定義される．ただし，pos はシーケンス上の位置，i は次元を表す．dim はモデルに入力する Embedding の次元数を表す．

$$PE_{(pos,i)} = \sin\left(pos/10000^{2i/dim}\right) \quad (1)$$

$$PE_{(pos,i+dim/2)} = \cos\left(pos/10000^{2i/dim}\right) \quad (2)$$

縦軸をシーケンスの位置，横軸を次元として，式 1，2 の定義によるパラメータをヒートマップで図 4 に表す．図 4 から視覚的に色の変化の小さい箇所を捉えることができる．シーケンス上の位置 (縦軸) の違いによる値の変化が小さい．すなわち，Position Embedding の一部の次元に含まれる情報量は小さいと言える．さらに，提案手法の Embedding は話者交代情報または話者識別情報をモデルに認識させるものであり，大きな次元数は必要ないと考えられる．我々は Position Embedding の情報量の小さい部分に限定して Turn/Speaker Embedding を加算することで，話者情報を効果的に認識させることができると考える．このように，全体の Embedding の次元数に対して小さい次元数の Turn/Speaker Embedding を Partial Turn/Speaker Embedding と呼ぶ．

本研究では最大シーケンス長を 512，Embedding

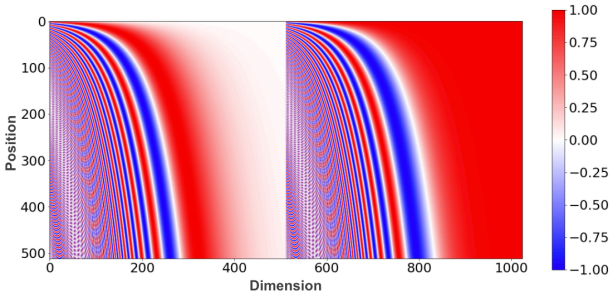

図 4　Sinusoidal Positional Embedding のパラメータのヒートマップ

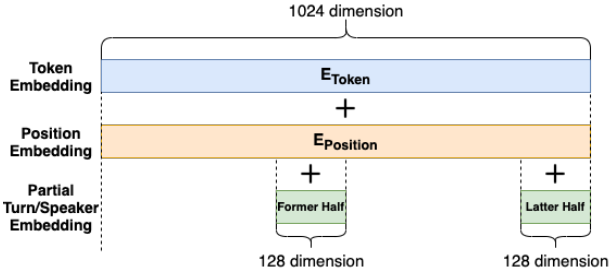

図 5　Partial Turn/Speaker Embedding のアーキテクチャ

の次元数を 1024 としているため，我々は 385〜512 次元と 897〜1024 次元を Position Embedding の情報量の小さい部分と考え，合計 256 次元の Turn/Speaker Embedding を構成する．本研究の設定を例に，Partial Turn/Speaker Embedding のアーキテクチャを図 5 に示す．本研究では PEGASUS に図 5 に示した方式で Partial Turn/Speaker Embedding を追加し，その影響を実験的に分析する．

4　実験手法

機械学習フレームワークとしては PyTorch[11] と HuggingFace Transformers[12] を，対話要約タスクのデータセットとして SAMSum[1] を使用した．要約モデルは PEGASUS[5] を用いて学習を行った．ベースラインの手法では HuggingFace[12] の公開する XSum[13] による事前学習済みの重みを初期パラメータとし，SAMSum を用いてファ

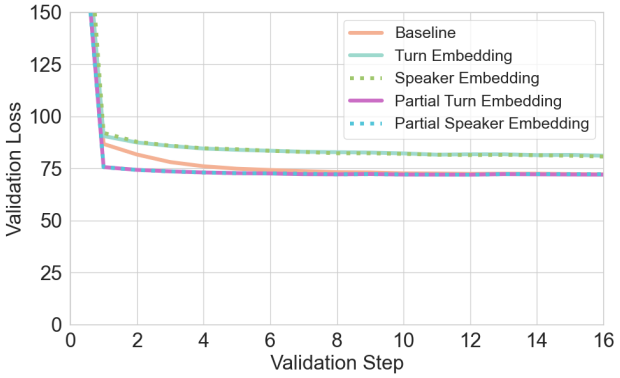

図 6　Validation Loss の変遷

インチューニングを行った．提案手法として 4 種類の実験を行った．ベースラインに対して Turn/Speaker Embedding を用いる場合の 2 手法と Partial Turn/Speaker Embedding を用いる場合の 2 手法を検証した．Position Embedding よりも Turn/Speaker Embedding の影響を大きくするため，提案手法に用いる追加の Embedding は 10 倍のスケーリングをした．評価指標には ROUGE[14] を用いた．さらに有意性の検証にランダム化 Tukey HSD 検定 [15] を用いた．

5　結果・考察

ベースラインと 4 種類の提案手法に対する Validation Loss の変遷と ROUGE スコアの比較を行う．

ベースラインと提案手法 4 種類の Validation Loss の変遷を図 6 に示す．Turn/Speaker Embedding を用いたときにはベースラインが辿り着いたような解に収束できていないことが見て取れる．しかし，Partial Turn/Speaker Embedding を用いた場合には同程度まで損失を下げつつ，収束性が向上していることがわかる．

ベースラインと提案手法 4 種類のテストデータにおける ROUGE-1 と ROUGE-2 に加えて，ROUGE-L のスコアを表 2 に示す．全体の Embedding と同様の次元数を持つ Turn/Speaker Embedding を用いた場合ではいずれの ROUGE スコアも大幅に下がってしまった．しかし，Partial Turn/Speaker Embedding を用いた手法では ROUGE スコアの向上が見られた．

さらに ROUGE スコアが向上した Partial Turn/Speaker Embedding を用いた手法に関して，ROUGE-2 のスコアに対してランダム化 Tukey HSD 検定 (サンプルサイズ $n = 819$，試行回数 $B = 5000$) を行い，それぞれの手法がベースラインと有意差を持つか検証を行った．その結果，ベースラインと Partial Turn Embedding を用いた手法による生成要約の ROUGE-2 の平均に有意差が認められた ($p = 0.011$)．さらに，ベースラインと Partial Speaker Embedding を用いた手法による生成要約の ROUGE-2 の平均にも有意差が認められた ($p = 0.0046$)．

表 2　テストデータにおける ROUGE スコア

Method	R-1	R-2	R-L
Baseline	50.68	26.54	43.04
Turn Embed	45.43	19.79	36.71
Speaker Embed	45.54	19.91	36.95
Partial Turn Embed	51.58	27.81	44.06
Partial Speaker Embed	**52.17**	**27.93**	**44.21**

6　結論・今後の課題

本研究では対話要約において話者交代情報または話者識別情報を含んだ Embedding を追加することによる効果を実験的に分析した．Embedding の次元を一部に限定して加算する Partial Turn/Speaker Embedding を用いることでモデルの収束性が向上し，すべての ROUGE スコアの向上も確認された．さらに，ベースラインと Partial Turn/Speaker Embedding を用いた手法の結果には，ランダム化 Tukey HSD 検定により ROUGE-2 において有意差が認められた．

今後の課題としては生成要約の人手評価や自動質問生成・自動質問応答システムを応用した QAGS[16] による評価を実施し，意味上の性能の違いを分析していきたい．また，追加の Embedding の初期化手法，スケーリングの設定をより細かく分析し，最良のパラメータを探索していきたい．さらに，話者情報を Embedding に包含させる手法は要約タスクに限らず，対話をドメインとするあらゆるタスクに応用できると考えられる．対話ドメインに限らず，ドメイン特有の情報を追加の Embedding の形式で与えることが可能かもしれない．同様の手法を対話ドメインの別のタスクや特殊なドメインのタスクに応用した場合の効果の分析もしていきたい．

参考文献

[1] Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, 2019.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin.

Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

[3] Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017.

[4] Yang Liu and Mirella Lapata. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019.

[5] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

[6] C. Goo and Y. Chen. Abstractive dialogue summarization with sentence-gated modeling optimized by dialogue acts. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, 2018.

[7] Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, Guillaume Lathoud, Mike Lincoln, Agnes Lisowska, Iain McCowan, Wilfried Post, Dennis Reidsma, and Pierre Wellner. The ami meeting corpus: A pre-announcement. In *Proceedings of the Second International Conference on Machine Learning for Multimodal Interaction*, 2005.

[8] Lin Yuan and Zhou Yu. Abstractive dialog summarization with semantic scaffolds, 2020. https://openreview.net/forum?id=B1eibJrtwr.

[9] Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

[10] Jia-Chen Gu, Tianda Li, Quan Liu, Zhen-Hua Ling, Zhiming Su, Si Wei, and Xiaodan Zhu. Speaker-aware bert for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, 2020.

[11] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, 2019.

[12] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020.

[13] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

[14] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, 2004.

[15] Benjamin Carterette. Multiple testing in statistical analysis of systems-based information retrieval experiments. *ACM Transactions on Information Systems - TOIS*, Vol. 30, No. 1, pp. 1–34, 2012.

[16] Alex Wang, Kyunghyun Cho, and Mike Lewis. Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.