質問応答によるメールからの送信者情報抽出

大田尾 匠　橋本 航
Sansan 株式会社
{sho.otao, wataru.hashimoto}@sansan.com

概要

本研究では、効率的なメール管理のために、メールから送信者情報 (氏名・会社名・部署・役職) を抽出する重要性に着目し、送信者のメールヘッダとメール本文を入力として、質問応答によって送信者情報を抽出する手法を提案する。実験では、Transformer をベースとした複数の言語モデルの性能を比較した。結果として、モデルのアーキテクチャによって抽出性能が高い項目が異なること、GPT-4o は正解が存在しない場合も誤抽出する傾向があるが Encoder-only モデルより性能が高いこと、入力長による性能低下はモデルや抽出項目によって違った傾向があることがわかった。

1　はじめに

メールから送信者情報 (氏名・会社名・部署・役職) を正確に抽出できれば、送受信履歴の解析や企業間関係の把握が容易になり、メール管理の効率化に繋がる。

送信者情報は、メール本文と送信者のメールヘッダ[1)]から抽出することができる。メールからの情報抽出の既存研究として、イベント (会議や依頼など) [1, 2] や署名 [3, 4, 5] を抽出する手法は提案されている。我々は、メール管理の効率化の観点から、メールの送信者情報に新しく着目する。また、テキストからの情報抽出手法の中で、近年では質問応答形式の有効性が示されている [6, 7, 8]。

本研究では、送信者のメールヘッダとメール本文の二つをプロンプトに含めて入力し、質問応答形式で送信者情報 (氏名・会社名・部署・役職) を抽出する手法を提案する。提案手法のイメージを図 1 に示した。実験では、Transformer [9] をベースとした複数のモデル (Encoder-only・Encoder-Decoder・Decoder-only) において性能を比較した。モデルは、長いメールを入力できるように入力可能系列長が 4096 トークン以上であり、かつ日本語に対応可能なものを用いた。実験の結果から、主に分かったことは以下である。

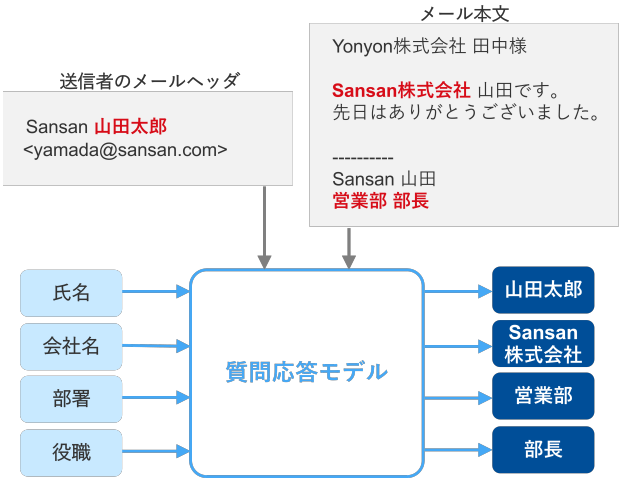

図 1: 提案手法

- Encoder-Decoder と Decoder-only のそれぞれの最良モデルを比べると、macro-f1 は同等だが、抽出性能が良い項目が異なった。
- GPT-4o は正解が存在しない場合も誤抽出する傾向があるが、Encoder-only モデルよりは性能が高くなった。
- 入力長の増加が必ずしも性能低下を引き起こすわけではなく、モデルや抽出項目によって違った傾向が見られた。

2　関連研究

2.1　メールからの情報抽出

メールから情報抽出を行う既存研究は数多く存在する。Patra ら [1] は会議情報を抽出する手法を提案した。Srivastava ら [2] は、会議や依頼など様々なイ

1)　本稿では、メールヘッダ「From: Sansan 山田太郎 <yamada@sansan.com> To: Yonyon 田中花子 <tanaka@yonyon.com> Subject: ...」の From の部分「Sansan 山田太郎 <yamada@sansan.com>」を送信者のメールヘッダと呼ぶ。

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

ベントを統一して扱い、メールから抽出する手法を提案した。また、Lampert ら [3] や Jardim ら [4] は、メール文から署名などのセグメントを抽出する手法を提案した。さらに、藤田ら [5] は対象を送信者に限定した署名抽出を行う手法を提案した。本研究では、効率的なメール管理の観点からメールの送信者情報に新しく着目し、抽出する手法を提案する。

2.2 質問応答による情報抽出

テキストからの情報抽出には従来、IOB タグ [10] を用いたトークン分類の手法が用いられてきたが、近年では Li ら [6, 7] や Lam ら [8] などによって、質問応答形式の有効性が示されている。また、本研究で取り組む送信者情報抽出では、同じ抽出項目に対して送信者のものかどうかを区別する必要がある。トークン分類では送信者のものかどうかを区別するためにタグが追加で必要となりタスクが複雑になるが、質問応答はその必要がなくシンプルである。これらの理由から、本研究では質問応答を用いた抽出手法を提案する。

2.3 長い系列長に対応した言語モデル

Transformer [9] ベースの言語モデルは、長い入力の処理に高い計算コストがかかるため、注意機構を工夫して効率よく計算を行う手法が数多く提案されている [11, 12, 13]。これらの手法により、メールのような長い入力も扱うことができる。一方で、入力を長くすると Decoder-only の推論性能が下がることが報告されている [14]。本研究では、メールの長さによって抽出性能が変化するかどうかを調べる。

3 実験

3.1 モデル

本研究では、送信者のメールヘッダとメール本文を入力して、送信者の情報 (氏名・会社名・部署・役職) を質問応答で抽出するモデルを提案する。モデルは、複数のアーキテクチャ (Encoder-only・Encoder-Decoder・Decoder-only) においてファインチューニングを行った。また、few-shot 推論の性能を確認するために、クローズドソースのモデルとして GPT-4o を用いて検証を行った。使用したモデルを以下に記載する。

Encoder-only　多言語に対応した XLM-RoBERTa [15] のチェックポイントを用いて、長い系列を入力可能にする Longformer [11] のスキームで事前学習をしたモデル[2)](以降、xlm-r-long) を検証に用いた。

Encoder-Decoder　T5 [16] をベースとした、長い系列を入力できる LongT5 [13] を多言語対応させた mLongT5 [17] の事前学習済みモデル[3)4)](以降、mlong-t5) を検証に用いた。

Decoder-only　Swallow LLM[5)]が公開している、日本語タスクにおける LLM 評価[6)]において、JSQuAD [18] の性能が高い事前学習済みモデル llm-jp-3[7)8)]、Qwen2.5[9)10)] 、gemma-2[11)] を検証に用いた。全てのモデルの入力長が 4096 トークン以上であり、長い系列にも対応できるモデルである。

GPT-4o　クローズドソースのモデルとして、Azure OpenAI Service[12)]が提供している GPT-4o の API (2023-03-15-preview) を検証に用いた。

3.2 実験設定

3.2.1 データセット

データセットは、Sansan 株式会社の社員が受け取ったメールのうち、メール本文が 4000 文字以下の 4787 件を使用した (平均長 1982 文字、詳細は付録 A.3 参照)。学習データとして 3740 件、検証データ 534 件、テストデータ 513 件を用い、検証およびテストデータには、学習データよりも過去のメールや、学習データと同じ氏名と会社名のペアを持つ人物が送信者のメールを含まないようにした。

各項目の正解は複数の表現を持つことがあるが、より詳細な情報を正解とした。例えば図 1 のように、”山田”と”山田太郎”が含まれる場合は”山田太郎”を氏名の正解とし、”Sansan”と”Sansan 株式会社”が含まれる場合は”Sansan 株式会社”を会社名の正解とする。また、項目ごとに正解が含まれる割合が異なり、氏名と会社名はほぼ全てのメールに記載され

2) https://huggingface.co/markussagen/xlm-roberta-longformer-base-4096
3) https://huggingface.co/agemagician/mlong-t5-tglobal-base
4) https://huggingface.co/agemagician/mlong-t5-tglobal-large
5) https://swallow-llm.github.io/index.ja.html
6) https://swallow-llm.github.io/evaluation/index.ja.html
7) https://huggingface.co/llm-jp/llm-jp-3-1.8b
8) https://huggingface.co/llm-jp/llm-jp-3-3.7b
9) https://huggingface.co/Qwen/Qwen2.5-1.5B
10) https://huggingface.co/Qwen/Qwen2.5-3B
11) https://huggingface.co/google/gemma-2-2b
12) https://learn.microsoft.com/ja-jp/azure/ai-services/openai/

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

表 1: 実験結果。各列で太文字は最も性能が高いモデル、下線付き文字は次に性能が高いモデルを表す。

model	archi.	氏名			会社名			部署			役職			全項目
		pre.	rec.	f1	pre.	rec.	f1	pre.	rec.	f1	pre.	rec.	f1	macro-f1
xlm-r-long (0.2B)	Encoder	0.937	0.929	0.933	0.864	0.843	0.853	0.618	0.770	0.686	0.058	0.250	0.094	0.642
mlong-t5 (0.5B)	Enc-Dec	0.962	0.962	0.962	0.920	0.921	0.920	0.821	0.862	0.841	0.230	0.107	0.146	0.717
mlong-t5 (1.2B)	Enc-Dec	0.966	0.966	0.966	0.931	0.933	0.932	0.915	**0.908**	**0.912**	0.312	0.178	**0.227**	**0.759**
llm-jp-3 (1.8B)	Decoder	0.976	0.976	0.976	0.933	0.935	0.934	0.914	0.850	0.881	0.125	0.071	0.090	0.720
llm-jp-3 (3.7B)	Decoder	**0.980**	**0.980**	**0.980**	**0.941**	**0.943**	**0.942**	**0.944**	0.845	0.892	0.230	0.107	0.146	0.740
Qwen2.5 (1.5B)	Decoder	0.964	0.964	0.964	0.910	0.912	0.911	0.907	0.862	0.884	0.142	0.071	0.095	0.713
Qwen2.5 (3.1B)	Decoder	0.978	0.976	0.977	0.933	0.933	0.933	0.928	0.862	0.894	**0.400**	0.071	0.121	0.731
gemma-2 (2.6B)	Decoder	0.970	0.970	0.970	**0.941**	**0.943**	**0.942**	0.898	0.887	0.893	0.240	0.214	0.226	0.758
GPT-4o (30-shot)	-	0.960	0.935	0.947	0.897	0.888	0.893	0.629	0.858	0.726	0.085	**0.821**	0.154	0.680

ているが、部署は約半数、役職は 1/10 以下となっている (詳細は付録 A.3 参照)。

3.2.2 学習

質問応答タスクとしては、1 つの質問ごとに 1 つの項目を抽出する形式とし、1 つのモデルに 4 つの項目抽出を学習させた。モデルへの入力は、項目抽出の指示文と、送信者メールヘッダとメール本文を適切に結合したプロンプト (付録 A.1) である。1 つのメールに対し、4 項目を抽出するプロンプトがそれぞれ存在するため、$3740 \times 4 = 14960$ 件を実際の学習データとして用いた。質問応答の形式は、Encoder-only は正解に該当するスパンを出力するように、その他のモデルは正解の文字列をそのまま出力するようにした。比較的モデルが大きい Decoder-only は LoRA [19] を用いたファインチューニングを行い、Encoder-only、Encoder-Decoder はフルパラメータチューニングを行った。クローズドソースの GPT-4o は、学習データを使って few-shot 推論を行った。各モデルのハイパーパラメータ設定は付録 A.2 に示す。

3.2.3 性能比較

性能は、スペースと改行文字を無視した完全一致で評価する。ただし、”山田 太郎”と”Yamada Taro”のような日本語表記と英語表記は同一視しない。モデルが何かを出力した数を N_{output} とし、正解が送信者メールヘッダまたはメール本文に存在している数を N_{exist} とする。また、正解が存在する場合に正解とモデルの出力が一致した数を N_{tp} とする。このとき、評価指標を以下のように定義する。

- precision = N_{tp}/N_{output}
- recall = N_{tp}/N_{exist}
- f1 = $2 * \text{precision} * \text{recall}/(\text{precision} + \text{recall})$

precision はモデルが出力をした中で正解している割合であり、recall は正解が存在している中でモデルが正しく出力できた割合である。項目によって正解が存在する割合が異なるため、各項目を公平に扱えるように、モデル全体の性能評価は各項目の f1 を平均した macro-f1 を用いて行う。

3.3 評価結果

テストデータに対する実験結果を表 1 に示す。最も高い macro-f1 を達成したのは、ほぼ同等の性能を示した Encoder-Decoder モデルの mlong-t5 (1.2B) と Decoder-only モデルの gemma-2 (2.6B) であった。mlong-t5 (1.2B) はパラメータ数は半分以下であるが、gemma-2 (2.6B) に匹敵する性能を達成している。macro-f1 は同等だが、両モデルの項目ごとの性能には違いが見られたことから、重視する項目に応じてモデルを選ぶことが重要である。また、役職は正解の存在割合が極端に少なく学習が難しいため、他項目と比べて抽出性能が低くなることがわかった。役職が存在しないメールにおいて人工的に役職を付与するなど、役職の正解が存在する割合を学習データで増やした場合の性能検証は今後の課題とする。

mlong-t5、llm-jp-3、Qwen2.5 においては同じモデルでパラメータ数を増やすと各項目の f1 が改善した。送信者情報抽出においては、同じモデルであればリソースが許す限りパラメータ数を大きくすることが望ましい。

few-shot 推論を行った GPT-4o は、Encoder-Decoder と Decoder-only の各モデルより性能は低いが、Encoder-only の xlm-r-long (0.2B) よりは性能が高くなった。xlm-r-long (0.2B) は、正解を予測スパンに含むがスパンの開始や終了の位置がずれるミスが目立ち、完全一致の評価を行う上で GPT-4o よりも性能が低くなった。また GPT-4o は、Encoder-Decoder や

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

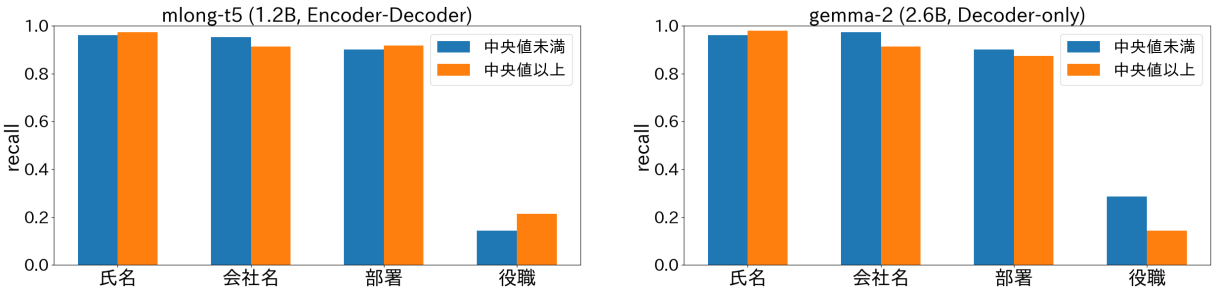

図 2: mlong-t5 (1.2B, Encoder-Decoder) と gemma-2 (2.6B, Decoder-only) における、プロンプトの長さによる recall の変化。正解が存在するデータを入力長の中央値で分割して、recall をそれぞれ計算した。

Decoder-only と比べて、部署と役職の precision が低いという特徴がある。これは、部署や役職の正解が存在しない場合でも、誤って何かを出力するケースが多いためである。Encoder-Decoder や Decoder-only では、部署と役職において、正解が存在しない場合は何も出力しない割合が高くなり、GPT-4o と比べて precision が改善した。他モデルと比べて GPT-4o の役職の recall が高い理由は、誤りを出力する数が増えると同時に、本来は正解が存在するが他モデルが何も出力しなかったメールに対して、正解を出力する数も増えるためである。

3.4 入力長による性能変化

入力が長くなった場合に、recall がどう変化するか分析した。検証には、macro-f1 が高い mlong-t5 (1.2B) と gemma-2 (2.6B) を用いた。各項目の評価対象は、テストデータのうち正解が存在するデータのみとした (件数は付録 A.3 の表 4 参照)。各項目ごとに、評価対象データの入力プロンプト長の中央値を基準にデータを二つのグループに分け、各グループについて recall を計算した。比較件数を揃えるために中央値で分割しており、例えば会社名は 256 件ずつ、役職は 14 件ずつのグループに分割した。

各項目の recall を計算した結果を図 2 に示す。Decoder-only の gemma-2 (2.6B) は、入力を長くすると氏名以外の項目で recall が低くなり、既存研究 [14] と同様の傾向が見られた。一方、Encoder-Decoder の mlong-t5 (1.2B) では、部署と役職において入力が長い方が recall が高くなっており、Decoder-only とは違う傾向があることが分かった。また、両モデルに共通して氏名は入力が長い方が少し recall が高くなっており、入力長以外にもメールの構造や送信者情報の位置などが抽出難易度に影響を与えていると考えられる。

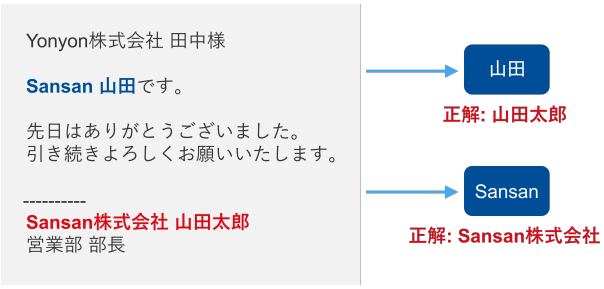

図 3: ミスケース

3.5 エラー分析

署名内に送信者のより詳細な情報が記載されているにもかかわらず、メール文中の他部分から出力しているミスケースが見られた (図 3)。このミスは、例えば藤田らの手法 [5] を使って送信者の署名を別で抽出し、我々の手法の抽出結果を送信者署名を用いて補完すると解決するかもしれない。このように、我々の提案手法と既存手法である署名抽出の両方を使った性能改善は今後の課題とする。

4 おわりに

本研究では、メールの送信者情報に着目し、送信者メールヘッダとメール本文を入力し、質問応答形式で送信者情報を抽出する手法を提案した。実験では、モデルによって抽出性能が高い項目が異なることや、GPT-4o は正解が存在しない場合にも誤抽出する傾向があるが Encoder-only モデルより性能が高くなることが分かった。また、入力長の増加が必ずしも性能低下を引き起こすわけではないことも分かった。今後の課題は、学習データにおいて役職の正解が存在するメールの数を増やした場合の性能検証と、署名抽出の既存手法を用いた補完によって特定のミスケースを解決できるかの検証である。

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

参考文献

[1] Barun Patra, Vishwas Suryanarayanan, Chala Fufa, Pamela Bhattacharya, and Charles Lee. ScopeIt: Scoping task relevant sentences in documents. In **Proceedings of the 28th International Conference on Computational Linguistics: Industry Track**, 2020.

[2] Saurabh Srivastava, Gaurav Singh, Shou Matsumoto, Ali Raz, Paulo Costa, Joshua Poore, and Ziyu Yao. MailEx: Email event and argument extraction. In **Proceedings of the Conference on Empirical Methods in Natural Language Processing**, 2023.

[3] Andrew Lampert, Robert Dale, and Cécile Paris. Segmenting email message text into zones. In **Proceedings of the Conference on Empirical Methods in Natural Language Processing**, 2009.

[4] Bruno Jardim, Ricardo Rei, and Mariana S. C. Almeida. Multilingual email zoning. In **Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop**, 2021.

[5] 藤田正悟, 高橋寛治. メールにおける送信者署名の抽出. Technical report, 情報処理学会, 2022.

[6] Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. Entity-relation extraction as multi-turn question answering. In **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, 2019.

[7] Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. A unified MRC framework for named entity recognition. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, 2020.

[8] Laurent Lam, Pirashanth Ratnamogan, Joël Tang, William Vanhuffel, and Fabien Caspani. Information extraction from documents: Question answering vs token classification in real-world setups. In **Document Analysis and Recognition - ICDAR**, 2023.

[9] 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 **Advances in Neural Information Processing Systems**, 2017.

[10] Lance Ramshaw and Mitch Marcus. Text chunking using transformation-based learning. In **Third Workshop on Very Large Corpora**, 1995.

[11] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. **arXiv:2004.05150**, 2020.

[12] Joshua Ainslie, Santiago Ontanon, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. ETC: Encoding long and structured inputs in transformers. In **Proceedings of the Conference on Empirical Methods in Natural Language Processing**, 2020.

[13] Mandy Guo, Joshua Ainslie, David Uthus, Santiago Ontanon, Jianmo Ni, Yun-Hsuan Sung, and Yinfei Yang. LongT5: Efficient text-to-text transformer for long sequences. In **Findings of the Association for Computational Linguistics: NAACL**, 2022.

[14] Mosh Levy, Alon Jacoby, and Yoav Goldberg. Same task, more tokens: the impact of input length on the reasoning performance of large language models. In **Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics**, 2024.

[15] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, 2020.

[16] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. **Journal of Machine Learning Research**, Vol. 21, No. 140, pp. 1–67, 2020.

[17] David Uthus, Santiago Ontanon, Joshua Ainslie, and Mandy Guo. mLongT5: A multilingual and efficient text-to-text transformer for longer sequences. In **Findings of the Association for Computational Linguistics: EMNLP**, 2023.

[18] Kentaro Kurihara, Daisuke Kawahara, and Tomohide Shibata. JGLUE: Japanese general language understanding evaluation. In **Proceedings of the Thirteenth Language Resources and Evaluation Conference**, 2022.

[19] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In **International Conference on Learning Representations**, 2022.

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

A 付録

A.1 プロンプト例

```
### 指示:
送信者のメールヘッダとメール本文を使って、メール送信者の氏名を応答に出力してください。

### 送信者のメールヘッダ:
{sender_email}

### メール本文:
{email_text}

### 応答:
{answer}
```

図 4: ファインチューニングに用いたプロンプトの例 (氏名)

```
送信者のメールヘッダとメール本文を使って、メール送信者の氏名を応答に出力してください。
以下の制約に従って出力してください。

- 送信者のメールヘッダもしくはメール本文に存在する文字列のみ出力してください。
- 送信者のメールヘッダもしくはメール本文にメール送信者の氏名が存在しない場合は、空白を出力してください。
- 他人の情報は含めず、送信者の情報のみを出力してください。
- 送信者の会社名、部署、役職は出力せず、送信者の氏名だけを出力してください。
```

図 5: GPT-4o の検証に用いた system プロンプトの例 (氏名)

A.2 ハイパーパラメータ

表 2: 各モデルのハイパーパラメータ。GPT-4o 以外のモデルでは複数 GPU で並列学習を行った。学習エポック数と GPT-4o に与える例の件数は、それぞれ検証データに対する macro-f1 が最も高くなる値を選択した。

	Encoder-only Encoder-Decoder	Decoder-only	GPT-4o
#GPU (A10, 24GB)	8		-
batch size	1		-
gradient accumulation steps	2		-
total batch size	16		-
epoch	{1,2,3,4,5}		-
optimizer	AdamW		-
precision	bf16		-
learning rate	$3e-5$		-
lora α	-	32	-
lora r	-	8	-
#example of few-shot	-		{0,10,20,30}

A.3 データセット

表 3: データセットに含まれるメール本文の長さ

項目	文字数
平均値	1982
中央値	1890
最小値	7
最大値	4000

表 4: 実験に用いたデータセット全 4787 件 (学習/検証/テスト=3740/534/513) において、送信者メールヘッダもしくはメール本文に正解が存在する件数

項目	正解が存在する件数 (学習/検証/テスト)
氏名	4779 (3732/534/513)
会社名	4777 (3731/534/512)
部署	2107 (1633/234/240)
役職	350 (301/21/28)

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).