メールからの送信者会社名抽出に特化した報酬設計に基づく Group Relative Policy Optimization

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

概要

メールから送信者の会社名を抽出するタスクでは、メール特有の文脈の複雑さや表記揺れにより、従来の Supervised Fine-Tuning（SFT）には性能の限界があった。本研究では、法人格やメールアドレスなどのメタ情報により出力を自動検証できる点に着目し、検証可能な報酬を複数設計して Group Relative Policy Optimization による追加学習を行った。実験の結果、法人格を考慮した部分一致報酬が最も有効であり、SFT を上回る抽出性能を達成した。さらに 1B 規模の軽量モデルでは、思考プロセスを明示する設定より最終出力のみを生成する設定の方が、推論コストと抽出性能の両面で優位であることを示した。

1　はじめに

メールからの送信者情報抽出は、膨大に蓄積された送受信履歴を効率的に利用できる基盤となる。特に会社名は、送信者を特定の組織に紐づけるための重要情報である。送信者情報はメール本文や送信者のメールヘッダから抽出でき、大田尾ら [1] はこれらをプロンプトに与え、Supervised Fine-Tuning(SFT) によって質問応答形式で送信者情報を抽出する手法を提案した。しかし、メール内には複数人の情報が混在することや、会社名の表記揺れ（法人格の有無や略称）などドメイン特有の課題があり、SFT のみでは性能改善の余地が残っていた。

近年、検証可能な報酬に基づく強化学習（Reinforcement Learning with Verifiable Rewards）[2, 3, 4] を用いて、言語モデルの推論能力を強化する手法が注目されている。その代表例である Group Relative Policy Optimization (GRPO)[2] は、同一プロンプトから生成された複数の出力をグループ内で相対的に評価し、モデルを更新する手法である。メールからの送信者会社名抽出は、正誤を自動で検証しやすい情報抽出タスクであり、法人格やメールアドレスといったメタ情報など活用可能な候補が多く報酬設計の探索余地も大きいことから、GRPO を用いることで SFT のみでは課題となっていた部分を改善できる可能性がある。

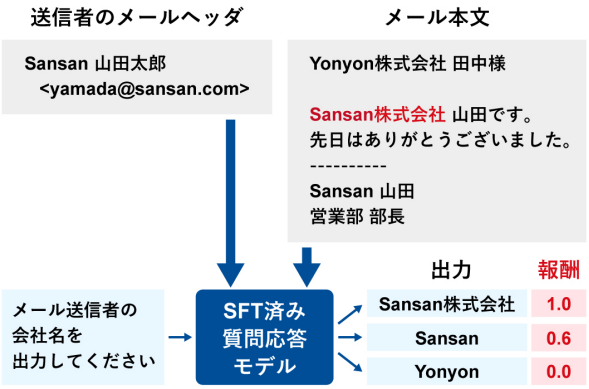

図 1: メールからの送信者会社名抽出における GRPO

また、GRPO の先行研究では Chain-of-Thought（CoT）[5] による思考プロセスをタグ内で明示させる設定が主流であり、数学・コーディング・科学推論などで高い性能が報告されている [2, 3]。しかし会社名抽出のように、最終的に短い文字列を得ることが目的のタスクでは、長い思考プロセスの生成は推論コストが増え、実運用における制約となる。Li ら [6] によって情報抽出への GRPO の適用例も報告されているが、思考プロセスを出力する設定である。

本研究では、メールからの送信者会社名抽出の性能向上のため、情報抽出タスクやドメインに特化した検証可能な報酬を設計し、GRPO による追加学習を行う手法を提案する (図 1)。さらに実運用を想定し、思考プロセスを明示する設定と最終出力のみを生成する設定を、抽出性能と推論コストの両面から比較評価する。 本研究の主な貢献は以下の通りである。

- 法人格を除いた部分一致報酬を用いた GRPO により SFT を上回る抽出性能を達成した。
- 1B 規模の軽量モデルにおける GRPO では、思考プロセスを明示するより最終出力のみを生成する方が、推論コスト・抽出性能の両面で優位であることを示した。

2 実験設定

2.1 Group Relative Policy Optimization

Group Relative Policy Optimization(GRPO)[2] は、以下の目的関数 $J(\theta)$ を最大化するように学習される。

$$J(\theta) = \mathbb{E}_{q\sim D,\ \{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}} \frac{1}{G}\sum_{i=1}^{G}\left\{ \min\left[r_i(\theta)A_i,\right.\right.$$
$$\left.\left.\text{clip}\Big(r_i(\theta), 1-\epsilon, 1+\epsilon\Big)A_i\right] - \beta\,\mathbb{D}_{KL}\big[\pi_\theta \,\|\, \pi_{\text{ref}}\big]\right\}.$$

ここで、D は学習データセット、$\{o_i\}_{i=1}^{G}$ は更新前の言語モデル $\pi_{\theta_{\text{old}}}$ によって同一プロンプトから生成された G 個の出力、π_θ は更新後の言語モデル、π_{ref} は学習前の参照モデル、$r_i(\theta)$ は更新前後の確率比 $\pi_\theta(o_i|q)/\pi_{\theta_{old}}(o_i|q)$、$\epsilon$ は確率比に対するクリッピング係数、β は更新後モデルと参照モデルの乖離を防ぐ KL ダイバージェンス正則化項の係数である。A_i は出力 o_i に対する、グループ内で標準化されたアドバンテージであり、R_i を報酬、$\mu_R(q)$ および $\sigma_R(q)$ をグループ内報酬の平均と標準偏差としたとき、$A_i = (R_i - \mu_R(q))/\sigma_R(q)$ と計算される。

2.2 学習設定

SFT と GRPO の性能を比較するために、まず事前学習済みモデルに対して SFT の学習を行う。次に、SFT 済みモデルに対して別データで SFT、GRPO の追加学習を行う。思考プロセスを伴わないモデルの学習・推論に用いるプロンプトは付録の図 2 に示す。思考プロセスを伴うモデルの学習・推論には、<thinking></thinking>タグの中に思考プロセスを記載し、<answer></answer>タグの中に正解を記載する (付録の図 3)。言語モデルは、日本語に対応しており、かつ 8192 トークン以上が入力可能な軽量事前学習済みモデルである、Qwen3(0.6B)[7][1] と sarashina2.2(1B)[2] を用いて実験を行った。また、SFT、GRPO ともに LoRA[8] で学習を行った。ハイパーパラメータは付録 A.2 に示す。

2.3 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 8000 文字以下の 9199 件を使用した (平均 2549 文字)。ベースとする SFT の学習データとして 3686 件、SFT や GRPO の追加学習に使うデータとして 3649 件、報酬の探索を行う検証データとして 928 件、テストデータとして 936 件に分割した。学習データと同じ会社名を持つ人物が、検証やテストデータの送信者に含まれないようにした。

思考プロセスを用いた SFT のためには、思考プロセスの正解データが必要である。本研究では、Azure OpenAI Service[3] が提供している GPT-4.1(2025-04-14) を用いて、入力と正解を与えたときに、正解を導き出す思考プロセスを生成した。プロンプトは付録の図 4 に示す。

2.4 評価指標

各項目の正解はより詳細な情報を正解とした。例えば図 1 のように「Sansan」と「Sansan 株式会社」が入力に含まれる場合は後者を正解とする。出力と正解の比較は、空白と改行文字を除去し、アルファベットと数字の全角半角を無視した完全一致とした。メール中に正解が存在しない場合は、空文字ではなく「不明」を正解とした。思考プロセスを伴う設定では<answer>タグの中身を出力とし、<answer>タグを正しく出力できていない場合には出力会社名は空文字とした。

本研究における評価指標は、正解が存在しない場合を考慮した指標とする。具体的には、モデルが「不明」以外を出力した数を N_{output}、正解が「不明」以外、つまり入力中に正解が存在している数を N_{exist}、「不明」以外で出力と正解が一致した数を N_{tp} とする。このとき、評価指標を以下のように定義する。

$$\text{Prec} = \frac{N_{tp}}{N_{output}},\quad \text{Rec} = \frac{N_{tp}}{N_{exist}},\quad \text{F1} = \frac{2 * \text{Prec} * \text{Rec}}{\text{Prec} + \text{Rec}}$$

Prec は「不明」以外の出力に対する正解の割合、Rec は正解が存在する入力に対する正解の割合である。

1） https://huggingface.co/Qwen/Qwen3-0.6B-base
2） https://huggingface.co/sbintuitions/sarashina2.2-1b
3） https://learn.microsoft.com/ja-jp/azure/ai-services/openai/

3 実験

3.1 報酬の定義

GRPO の学習に用いる報酬を定義する。複数報酬を重み付き加算したものを最終報酬とするため、全ての報酬のスケールを $[0, 1]$ に揃えている。出力と正解の文字列一致を評価する R_{partial} を情報抽出のベースラインとし、思考プロセスを含む場合の報酬として R_{format} を用いる。ドメインに特化した報酬として、R_{exist}、R_{marker}、R_{domain} を提案する。それぞれの定義を以下に述べる。

部分一致 (R_{partial}) 情報抽出タスクの根幹をなす報酬は以下のように計算する。

$$\frac{2 * (\text{出力と正解の最長共通部分文字列の長さ})}{(\text{出力文字列の長さ}) + (\text{正解文字列の長さ})}$$

ここで最長共通部分文字列とは、二つの文字列で連続して一致する最長の文字列を指す。正解に近い出力を適切に評価するため、モデルの出力が正解と完全一致していれば 1.0、部分一致していれば部分点を与える。また、フォーマット一致報酬 (R_{format}) と併用する際は、<answer></answer>に囲まれた文字列を出力として扱う。

フォーマット一致 (R_{format}) 思考プロセスを伴うモデルの学習のため、出力が思考フォーマットに準拠しているかを評価する報酬を導入する。具体的には、出力に<thinking>および</thinking>タグで囲まれた文字列が存在し、かつその直後に<answer>および</answer>で囲まれた文字列が存在していれば 1.0、そうでなければ 0.0 の報酬を与える。

存在性 (R_{exist}) 情報抽出タスクでは、正解が文書中に記述されている場合は正解が必ず入力に含まれるので、入力に存在しない文字列の出力を抑制する報酬を導入する。具体的には、モデルの出力が入力中に含まれていれば 1.0、含まれていなければ 0.0 を与える。ただし、正解が文書中に存在せず「不明」と出力している場合は報酬は 1.0 を与える。

法人格を除いた部分一致 (R_{marker}) 会社名には「株式会社」等の法人格が含まれ、会社名部分の本質的な一致判定を妨げる場合がある。そこで、付録の図 5 に記載した 19 個の法人格を出力と正解から削除したうえで R_{partial} を計算する。

送信者メールドメインとの一致率 (R_{domain}) 本文中に複数の会社名が混在する場合に、送信者自身の会社名を特定するため、メールアドレスのドメイン[4]を活用した報酬を導入する。報酬の算出手順は以下の通りである。

1. 会社名の表記揺れに対応するため、Sudachi 同義語辞書 [9][5] および chikkarpy パッケージ[6] を用いて、出力会社名の同義語リストを生成する。
2. リストの会社名を pykakasi パッケージ[7] を用いてローマ字に変換する。
3. ローマ字化した各会社名と、送信者メールアドレスのドメイン文字列との間で R_{partial} を計算し、その最大値を報酬値として採用する。

表 1: 思考無しの GRPO における報酬の探索。

報酬	報酬比率	Prec	Rec	F1
R_{partial}	-	0.929	0.929	0.929
$R_{\text{partial}} + R_{\text{exist}}$	20:80	0.920	0.920	0.920
	50:50	0.917	0.917	0.917
	80:20	0.927	0.927	0.927
$R_{\text{partial}} + R_{\text{marker}}$	20:80	**0.931**	**0.931**	**0.931**
	50:50	0.925	0.925	0.925
	80:20	0.922	0.922	0.922
$R_{\text{partial}} + R_{\text{domain}}$	20:80	0.009	0.009	0.009
	50:50	0.922	0.922	0.922
	80:20	0.928	0.928	0.928

3.2 GRPO における最適報酬の探索

第 3.1 節で定義した報酬を用い、会社名の抽出性能を最大化する最適な報酬構成を探索した。思考プロセス無しの設定で、まず SFT で学習を行い、次に SFT 済みモデルに対して追加学習データを用いて GRPO を学習した。モデルは Qwen3(0.6B) を用いた。評価対象として、テストデータとは別の検証データ 928 件を用いた。基本指標である部分一致報酬 R_{partial} をベースラインとし、特定のドメイン特化報酬を重み付き加算（比率 20:80, 50:50, 80:20）で組み合わせた際の性能変化を検証した。報酬や報酬比率の組み合わせの網羅的検証については、今後の課題とする。

実験の結果を表 1 に示す。R_{partial} と R_{marker} の組み合わせが最も高い性能を示した。これは、会社名における法人格の有無の考慮を報酬に反映させることが有効であることを示唆している。一方で、R_{exist} と R_{domain} は性能改善には大きく寄与しなかった。特に、R_{domain} の比率を高めた際、全ての出力がメール

4) @とドットの間の文字列をドメインと定義する。例えば「@sansan.com」だと「sansan」がドメインである。
5) https://github.com/WorksApplications/SudachiDict/
6) https://pypi.org/project/chikkarpy/
7) https://pypi.org/project/pykakasi/

表 2: SFT 済みモデルに対して、SFT や GRPO の追加学習を行ったモデルにおける性能比較。

学習方法	思考	Qwen3			sarashina2.2		
		Prec	Rec	F1	Prec	Rec	F1
SFT	無	0.912	0.914	0.913	0.923	0.925	0.924
SFT→SFT	無	0.923	0.925	0.924	0.933	0.935	0.934
SFT→GRPO*	無	**0.926**	**0.928**	**0.927**	**0.937**	**0.939**	**0.938**
SFT	有	0.832	0.834	0.833	0.849	0.851	0.850
SFT→SFT	有	0.872	0.874	0.873	0.877	0.879	0.878
SFT→GRPO†	有	0.925	0.927	0.926	0.923	0.925	0.924
SFT→GRPO‡	有	0.921	0.923	0.922	0.627	0.628	0.628

* : $R_{\text{partial}}(20) + R_{\text{marker}}(80)$
† : $R_{\text{format}}(50) + R_{\text{partial}}(10) + R_{\text{marker}}(40)$
‡ : $R_{\text{format}}(25) + R_{\text{partial}}(15) + R_{\text{marker}}(60)$

ドメインになり、性能が大きく低下した。R_{domain} の割合を減らしても R_{partial} のみで学習したモデルより性能が改善しなかったのは、正しい会社名でもメールドメインとの文字列一致率が低い例が多いためである。検証データの会社名の正解と送信者メールアドレスを用いて R_{domain} を計算すると、値が 0.5 未満と小さいケースが 47.8%存在していた。同義語辞書を用いても、メールドメイン特有の略称や表記揺れを完全には吸収できず、報酬としてのノイズとなったと考えられる。また、R_{partial} のみを報酬として用いて GRPO を学習したモデルにおいて、入力プロンプト外の文字列を出力する事例は検証データ全 928 件中、3 件と稀であった。この結果から、0.6B という軽量なモデルにおいても、指示に従った抽出能力は概ね備わっており、R_{exist} を用いることで報酬としてのノイズとなったと考えられる。

3.3 SFT と GRPO における抽出性能比較

SFT 済みモデルに対し、追加学習データを用いて SFT および GRPO を適用し、テストデータで性能を比較した。GRPO の報酬は、第 3.2 節で特定した最適構成を採用した。また、思考プロセスを明示的に出力させた場合の実験も行い、GRPO の報酬には第 3.2 節で選定した R_{partial} および R_{marker} を 1:4 で組み合わせるのに加え、フォーマット報酬 R_{format} を導入し、R_{format} の比率を変えて実験した。

結果を表 2 に示す。思考プロセス無しの結果として、両モデルで GRPO が SFT を上回る性能を達成した。これは、ドメイン知識を用いた適切な報酬設計の下では、同一の追加学習データを用いても SFT より GRPO が有効に機能することを示している。

思考プロセス有の設定では R_{format} の比率を小さくすると性能が低下したため、形式維持の報酬が重要

表 3: SFT 済みモデルに対して GRPO の追加学習を行ったモデルにおける、思考出力の有無による推論コスト比較。#Token と Time[s] はテストデータにおける平均出力トークン数と平均推論時間を表す。

学習方法	思考	Qwen3		sarashina2.2	
		#Token	Time[s]	#Token	Time[s]
SFT→GRPO*	無	8.79	0.27	4.11	0.11
SFT→GRPO†	有	172.35	4.79	139.71	2.72

* : $R_{\text{partial}}(20) + R_{\text{marker}}(80)$
† : $R_{\text{format}}(50) + R_{\text{partial}}(10) + R_{\text{marker}}(40)$

とわかる。また、思考プロセス有の設定においても GRPO は SFT よりも高性能だったが、思考プロセス無しの GRPO の性能には及ばなかった。この要因としては、情報抽出では CoT の寄与が限定的であること [10] や、1B 規模の軽量モデルでは複雑な思考を学習することが難しい可能性が考えられる。

3.4 思考の有無による推論コストの比較

GRPO で学習したモデルについて、思考プロセスが推論コストに与える影響を評価した。GPU(L40S) を 1 枚用いて、transformers パッケージ[8] の generate 関数でテストデータ 936 件を 1 件ずつ推論し、平均出力トークン数と平均推論時間を比較した。

結果を表 3 に示す。思考プロセスを出力しない設定は、両モデルにおいて出力トークン数が大幅に少なく、推論時間も短かった。第 3.3 節の結果も踏まえると、1B 規模の軽量モデルでは、思考プロセスを出力しない方が抽出性能と推論コストの両面で優位であることがわかった。より大規模なモデルにおける GRPO の検証は今後の課題とする。

4 おわりに

メールからの送信者会社名抽出において、メール特有の複雑な文脈や表記揺れといった課題があり、SFT では性能改善の余地があった。本研究では、メールからの送信者会社名抽出に特化した報酬を複数設計し、GRPO による追加学習を行う手法を提案した。実験の結果、法人格を考慮した部分一致を報酬に用いることで SFT を上回る精度を達成した。また、思考プロセスを明示させない方が、1B 規模の軽量モデルにおいて抽出性能および推論コストの両面で優れることを示した。今後の課題は、報酬や報酬比率の組み合わせの網羅的検証と、より大規模なモデルにおける GRPO の検証である。

8) https://pypi.org/project/transformers/

参考文献

[1] 大田尾匠, 橋本航. 質問応答によるメールからの送信者情報抽出. 言語処理学会第 31 回年次大会, 2025.

[2] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. **arXiv preprint arXiv:2402.03300**, 2024.

[3] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. **arXiv preprint arXiv:2501.12948**, 2025.

[4] Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1.5: Scaling reinforcement learning with llms. **arXiv preprint arXiv:2501.12599**, 2025.

[5] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. **Advances in neural information processing systems**, 2022.

[6] Ran Li, Shimin Di, Yuchen Liu, Chen Jing, Yu Qiu, and Lei Chen. Beyond path selection: Better llms for scientific information extraction with mimicsft and relevance and rule-induced (R ^{2}) grpo. **arXiv preprint arXiv:2505.22068**, 2025.

[7] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. **arXiv preprint arXiv:2505.09388**, 2025.

[8] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. **International Conference on Learning Representations**, 2022.

[9] Kazuma Takaoka, Sorami Hisamoto, Noriko Kawahara, Miho Sakamoto, Yoshitaka Uchida, and Yuji Matsumoto. Sudachi: a japanese tokenizer for business. In **Proceedings of the Eleventh International Conference on Language Resources and Evaluation**, 2018.

[10] Zayne Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning. **International Conference on Learning Representations**, 2025.

[11] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. **Advances in neural information processing systems**, 2025.

[12] Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. In **Conference on Language Modeling**, 2025.

[13] Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. **arXiv preprint arXiv:2503.24290**, 2025.

A 付録

A.1 プロンプト

```
### 指示:
送信者のメールヘッダとメール本文を使って、メール送信者の会社名を応答に出力してください。
送信者の会社名が記述されていない場合は、「不明」と出力してください。

### 送信者のメールヘッダ:
{sender_email}

### メール本文:
{email_text}

### 応答:
{answer}
```

図 2: 思考プロセスを伴わないモデルの学習・推論に用いたプロンプト。

```
### 指示:
送信者のメールヘッダとメール本文を使って、<thinking>と</thinking>の間に送信者の会社名を推論するための論理的な思考過程を記載し、<answer>と</answer>の間に送信者の会社名を出力してください。
送信者の会社名が記述されていない場合は、「不明」と出力してください。

### 送信者のメールヘッダ:
{sender_email}

### メール本文:
{email_text}

### 応答:
<thinking>{thinking}</thinking><answer>{answer}</answer>
```

図 3: 思考プロセスを伴うモデルの学習・推論に用いたプロンプト。

```
### 指示:
あなたはメールから送信者の会社名を抽出する専門家です。メール本文と送信者メールアドレスと送信者の会社名の正解が与えられます。メール本文と送信者メールアドレスから、与えられた送信者の会社名を推定する際の「ユーザー向けの論理的な思考」をstep-by-step で作成し、<thinking>と</thinking>タグで囲んで出力してください。その後に、与えられた正解の会社名を出力してください。送信者の会社名が記述されていない場合は、「不明」と出力してください。

### メール本文:
{email_text}

### 送信者のメールアドレス:
{sender_email}

### 送信者の正解:
{answer}

与えられた正解は絶対です。入力と違う正解を出力しないでください。
```

図 4: 思考プロセスの正解データを用意するために、GPT-4.1 に与えるプロンプト。

A.2 ハイパーパラメータ

SFT と GRPO の学習に使用したハイパーパラメータを表 4 に示す。GRPO の各学習ステップにおけるモデル更新回数 (num iterations) は、GRPO の提案論文 [2] に従い 1 に設定した。また、GRPO のいくつかの後続研究 [11, 12, 13] において KL ダイバージェンス正則化項が使われていないことを踏まえ、正則化項の係数 β は 0.0 に設定した。

表 4: SFT と GRPO の学習に使用したハイパーパラメータ。

	SFT	GRPO
#GPU (L40S)	8	
batch size	1	
gradient accumulation steps	2	8
total batch size	16	64
epoch	1	
precision	bf16	
learning rate	$3e^{-5}$	
lora α	32	
lora r	8	
lora dropout	0.05	
num generations	-	8
temperature	-	1.0
top p	-	1.0
clip ϵ	-	0.2
num iterations	-	1
KL β	-	0.0

A.3 報酬計算に用いる法人格

```
株式会社、有限会社、合同会社、合資会社、合名会社、相互会社、特殊会社、信用金庫、信用組合、信用保険会社、学校法人、社団法人、財団法人、医療法人、監査法人、国立大学法人、(株)、(有)、(合)
```

図 5: 法人格リスト