進捗報告

1　モデルマージのハルシネーションの原因調査

研究会のためにモデルマージの実験をした際に, 予備実験で出力が安定していた Elyza と Vicuna 同士で Merge したにもかかわらずチャットボットとして短文で応答せず出力トークンが max まで出力された例が見られた.

パラメータをマージすることにより起こるものなのかと思ったが, Elyza の出力確認に用いたテキスト生成のパラメータと Vicuna とモデルマージの際に用いたテキスト生成のパラメータが違うことが原因だと判明した. 表 1 に前者, 表 2 に後者のパラメータを示す. 表 2 のパラメータに変更した理由として, 先行研究 [1] で表 2 の temperture が　0.3 となったパラメータを用いていたからである.

表 1: Elyza の出力確認のために用いていたパラメータ

パラメータ名	値
do_sample	false
num$_b eams$	2
max_new_tokens	64

表 2: Elyza の出力確認のために用いていたパラメータ

パラメータ名	値
do_sample	True
temperature	0.3
max_new_tokens	512

実際に, FT 後の Elyza に対して表 2 のパラメータでテキスト生成をさせると, 出力トークン上限最大まで応答し続ける挙動が多く見られた. 以下は 「旅行に行くとしたら、どこがいい？」の応答である. このように同じ文言を繰り返す応答が他のクエリに対しても見られた. EOS トークンがうまく学習されていないことが考えられる.

Elyza は学習の際に以下のように Padding が EOS トークンで代用された関係で損失計算する部分の末尾に付けた EOS トークンがきちんとラベル付けされていない状態となっていた. この状態でも表 1 のパラメータの生成でうまくいっていたのでスルーしてしまっていたが, pad_token = eos_token とする処理が内部的にされている可能性があるので調査する. tokenizer.config では pad_token = null となっていた[1].

elyza/ELYZA-japanese-Llama-2-7b-instruct の場合

損失を計算する部分の元データ
こんにちは。お会いできて嬉しいですわ。
損失を計算する部分のラベル
13, 30589, 30389, 30353, 30644, 30449, 30267, 30697, 30437, 30298, 30499, 30538, 30466, 232, 175, 140, 30326, 30298, 30499, 30427, 31068, 30267

[1] https://huggingface.co/elyza/ELYZA-japanese-Llama-2-7b-instruct/blob/main/tokenizer_config.json

表 2 のパラメータで生成した Elyza の応答

ミュージカルの本場ロンドンですわ。ミュージカルが大好きですの。ミュージカルの本場ですの。ミュージカルが上演されている劇場に立ち会いたいですわ。ミュージカルの本場ですわ。ミュージカルが上演されている劇場に立ち

表 3 に示すモデルマージの結果においても, Elyza が Vicuna と比較してスコアが大幅に低かったのもこの応答が原因と考えられる.

表 3: 実験 2 結果

モデル	GPT-4o-mini の評価
Vicuna (FT)	4.92
Elyza (FT)	4.16
Merge	**5.49**

2 今後やること

先週の発表練習, 月曜の研究会で頂いた意見を踏まえ, 優先度高い順に記述します.

- RAG の新たなアプローチの検討
- Mergekit-evolve の具体的な進化計算

  README を見た感じ, CMA-ES というアルゴリズムを用いているらしい. 実装部分 [2] らしき箇所を見つけたがまだ理解しきれてない. 具体的にはどのように世代数を決定しているか, default で 100 とされている max-fevals のパラメータの意味の 2 つが理解できていない.
- Elyza の代替となる Llama2 派生の LLM 調査
- モデルマージの評価者 LLM 用のプロンプト作成 (応答としての自然さを評価基準に加える)
- 学習・推論時のプロンプトを英語にする

  現在は「お嬢様のようにふるまってください〜(以下略)」と ChatHaruhi [2] のデータセットにあったシステムプロンプトを日本語訳しているが, 英語のまま＋「必ず日本語で応答してください」として試してみる/
- OpenAI のインターフェース用いたファインチューニング試す
- 森先生のアイデアの RAG の実装

参考文献

[1] Yunfan Shao, Linyang Li, Junqi Dai, and Xipeng Qiu. Character-llm: A trainable agent for role-playing, 2023.

[2] Cheng Li, Ziang Leng, Chenxi Yan, Junyi Shen, Hao Wang, Weishi MI, Yaying Fei, Xiaoyang Feng, Song Yan, HaoSheng Wang, Linkang Zhan, Yaokai Jia, Pingyu Wu, and Haozhen Sun. Chathaviving anime large language model, 2023.

[2] https://github.com/arcee-ai/mergekit/blob/main/mergekit/scripts/evolve.py