＊JGLUEデータを学習データとして利用することは経産省の指示により禁止されたため、適宜、他のファインチューニング用データを設定してください。

開発アイデア①:

当チームでは、モデルの深さと幅の探索に焦点を当てた開発を行う。多くの先行開発されたLLMでは、一定のパラメータのもとでモデルの深さ(レイヤー数)と幅(ディメンション数)についてのコンセンサスが存在する。例えば、GPT-3, Llama 2, LLM-jpの13Bモデルはいずれもレイヤー数が40でディメンション数が5120~5140である[1]。しかし、これらの値の決定について根拠が公開されていない。当チームではこれらの値を独自に探索することで最適化を図る。具体的に、モデル構造はデコーダオンリーのトランスフォーマー構造(GPT系のモデル構造)とし、事前学習時に1M～1B程度の小規模モデルでレイヤー数とディメンション数の間の黄金比を見つけ出し[2]、10Bモデルの学習時にその黄金比を用いる。事前学習に用いるデータはJapanese mC4とWikipediaから構成される日本語データ、ファインチューニングに用いるデータは、JGLUEの訓練データおよび日本語訳されたDolly会話データを想定する[3]。余裕があればアテンションのヘッド数も追加で探索対象とする。

開発アイデア②:

当チームでは、モデル構造の変更に焦点を当てた開発を行う。トランスフォーマーにはモデル構造を決定するハイパーパラメータが数多く存在する。活性化関数、位置エンコーディング、残差接続、正則化がその例であるが、これらを調整して最適なモデル構造を見つけ出し、精度改善を行う。具体的に、モデル構造はGPT系のモデル構造とし、事前学習時に1M～1B程度の小規模モデルで最適なモデル構造を見つけ出し、10Bモデルの学習時にその構造を用いる。事前学習に用いるデータはJapanese mC4とWikipediaから構成される日本語データ、ファインチューニングに用いるデータは、JGLUEの訓練データおよび日本語訳されたDolly会話データを想定する。余裕があれば非トランスフォーマー構造(Mamba, RetNet, RWKV等)のモデルを実装してトランスフォーマーとの精度比較を行う。

開発アイデア③:

当チームでは、トークナイゼーションの改善に焦点を当てた開発を行う。一般的にトークナイゼーションは事前学習データの一部で実施するが、Googleが発表したGeminiのテクニカルレポートによると、学習データ全量でトークナイゼーションを行うと性能が改善したとの報告がある[4]。当チームもこれに倣い、トークナイゼーションの最適化によって精度改善を行う。具体的に、モデル構造はGPT系のモデル構造とし、事前学習データをできる限り多く利用してトークナイゼーションを行う。最適なボキャブラリサイズについても探索を行う。トークナイゼーションの最適化によって同じ計算量(FLOPS)で処理できるテキスト量の増加が期待されるため、事前学習に用いるデータはJapanese mC4とWikipediaから構成される日本語データの他にredpajama2からの英語データも適宜混ぜる。ファインチューニングに用いるデータは、JGLUEの訓練データおよび日本語訳されたDolly会話データを想定する。余裕があればファインチューニング時にも、ファインチューニングデータでトークナイゼーションを行って単語追加修正を行う。

開発アイデア④:

当チームでは、事前学習データの学習順序探索に焦点を当てた開発を行う。LLMにおける言語間の知識転移は、Llama2モデルを日本語データで継続学習するといった方法で、既に多くの先行開発された日本語LLMで効果が検証されている。継続学習のメリットは、少量の日本語データを大量の英語データで補完して日本語能力の改善に利用できることである。具体的に、モデル構造はGPT系のモデル構造とし、事前学習の最初は英語(Japanese mC4とWikipedia)とプログラム言語(The Stack)の混合データで学習し、その後、日本語データ(Japanese mC4とWikipedia)による継続学習を行う。ファインチューニングに用いるデータは、JGLUEの訓練データ

---

[1] https://arxiv.org/pdf/2302.13971.pdf, https://arxiv.org/pdf/2005.14165.pdf, https://huggingface.co/llm-jp/llm-jp-13b-v1.0

[2] 事前学習におけるValidation Lossを性能の代理変数として使用する。

[3] https://huggingface.co/datasets/databricks/databricks-dolly-15k

[4] https://assets.bwbx.io/documents/users/iqjWHBFdfxIU/r7G7RrtT6rnM/v0

および日本語訳されたDolly会話データを想定する。余裕があれば、限られた日本語データ量を補うために、疑似データを作成して混ぜる対応も行う。短い文章→長い文章の順番といったカリキュラム学習の効果を検証する。

開発アイデア⑤:
当チームでは、事前学習データの精度向上によるハルシネーションの逓減に焦点を当てた開発を行う。一般公開されている日本語学習データ(Japanese mC4)は英語データに比べるとデータ量は小さく、また品質も高くはない。ハルシネーションの原因の一つに学習データの品質問題があることは以前から指摘されている。そこで学習データの品質向上によって精度改善を試みる。具体的には、公開されている中でも大きい日本語データであるJapanese mC4について、refinewebの論文[5]で実施されているようなフィルタリングや重複排除処理による品質改善を実施する。ファインチューニングに用いるデータは、JGLUEの訓練データおよび日本語訳されたDolly会話データを想定する。余裕があれば安全性を高めるために、学習データ内の個人情報をマスクする対応も行う。

開発アイデア⑥:
当チームでは、ファインチューニングデータの組み合わせ探索に焦点を当てた開発を行う。具体的に、モデル構造はGPT系のモデル構造とし、ファインチューニングデータの種類(選択式問題、会話形式データ)、各データの配合比率、学習エポック数などを変化させて性能を改善する。ファインチューニングによる性能改善に焦点を当てるため、事前学習の探索は最小限に抑えて素早く完了させて、ファインチューニングの試行錯誤に多くの時間を充当する。事前学習に用いるデータはJapanese mC4とWikipediaから構成される日本語データ、ファインチューニングに用いるデータは、JGLUEの訓練データ[6]および日本語訳されたDolly会話データ[7]に加えて、日本語訳oasst1会話データセット[8]、llm-japanese-dataset[9]等を想定する。余裕があれば, Zero-shotの頑健性向上を目的としたインストラクション文を多様化させたファインチューニングや, RLHF(人間のフィードバックによる強化学習)もあわせて実施する。

開発アイデア⑦:
当チームでは、学習の最適化に焦点を当てた開発を行う。LLMのパラメータ学習時に設定するハイパーパラメータは数多く存在する。例えば、パラメータの初期化方式、Optimizerの種類、Weight Decay、学習エポック数である。これらハイパーパラメータの最適化によって精度改善を行う。具体的に、モデル構造はGPT系のモデル構造とし、事前学習時に1M～1B程度の小規模モデルで最適なハイパーパラメータを見つけ出し、10Bモデルの学習時にその値を用いる。事前学習に用いるデータはJapanese mC4とWikipediaから構成される日本語データ、ファインチューニングに用いるデータは、JGLUEの訓練データおよび日本語訳されたDolly会話データを想定する。余裕があれば事前学習時の目的関数の工夫、例えばUL2[10]に記載されたデノイジング学習を行う。

開発アイデア⑧:
当チームでは、学習の効率化・高速化に焦点を当てた開発を行う。事前学習での大規模データの学習にかかる計算量(FLOPS)と時間を改善することは開発コストを抑えるために重要なテーマである。具体的に、モデル構造はGPT系のモデル構造とし、Flash Attention[11]やSparse Attention, RMSNorm等の導入による学習の高速化を図る。学習の高速化が図れる分, 事前学習時に1M～1B程度の小規模モデルで広範囲のハイパーパラメータを探索し、最適値を見つけ

[5] https://arxiv.org/pdf/2306.01116.pdf
[6] https://huggingface.co/datasets/shunk031/JGLUE
[7] https://huggingface.co/datasets/kunishou/databricks-dolly-15k-ja
[8] https://huggingface.co/datasets/kunishou/oasst1-89k-ja
[9] https://github.com/masanorihirano/llm-japanese-dataset
[10] https://arxiv.org/pdf/2205.05131.pdf
[11] https://github.com/microsoft/Megatron-DeepSpeed/tree/7f7cea32bc7b13ec8b3981b1a4616ed5d5dc48a3/examples_deepspeed/rebase#flash-attention

る。事前学習に用いるデータはJapanese　mC4とWikipediaから構成される日本語データ、ファインチューニングに用いるデータは、JGLUEの訓練データおよび日本語訳されたDolly会話データを想定する。余裕があればMixture-of-Expert（MoE）化を行い、更なる学習の効率化を図る。