

研究発表

文献

背景

近年大規模言語モデル(LLM)の数が増加

- ChatGPT(OpenAI), LLaMa(Meta)
- LLMの中でも、大規模のものと小規模のものがある.

パラメータ

- 「パラメータ数」：言語モデルの規模を表す指標の一つ.
- モデルが学習する際に調整される変数のこと.

多様なLLM

- 大規模なLLM：計算コストやリソースが高いが、ハイスペック.
- 小規模なLLM：
モデルのサイズを抑えることができるが、高度な質問等の精度は低い.

問題

生成言語モデル多すぎ問題

- 簡単な質問に答えるのにハイスペックモデルを使うのは、リソースの無駄遣い.
- ユーザが簡単な質問をするか、複雑な質問をするかを予想するのは困難.

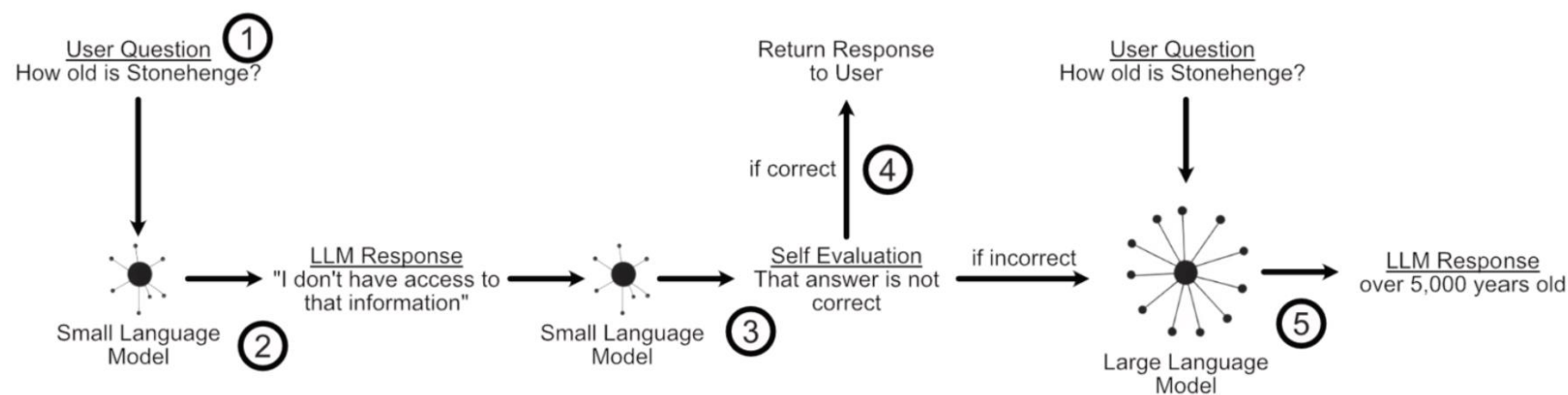
LLMルーティング

- 入力されたクエリを分析し、そのクエリに適したモデルをルーティングする.
- パフォーマンスを維持しながら、コストを削減する.

先行研究

LLMカスケード

- 小さいモデルと大きいモデルのどちらを使用するかを判別できる.
- AutoMix, FrugalGPT



(1) 小さな LLM にクエリを渡し、(2) 応答を生成します。(3) 次に、同じ小さな LLM に質問が正しく回答されたかどうかを尋ねます。(4) 正しく回答された場合は、結果を返します。(5) 小さな LLM が正しくなかった場合は、より大きな LLM を使用して再試行します。

課題点

- 数多くあるLLMからどのLLMを選んでくるかがの方法が確立されていない。

目的

大量にあるLLMから候補となるLLMを選ぶ

- 人がLLMの候補を選ぶより，機械的に候補を選別できる.

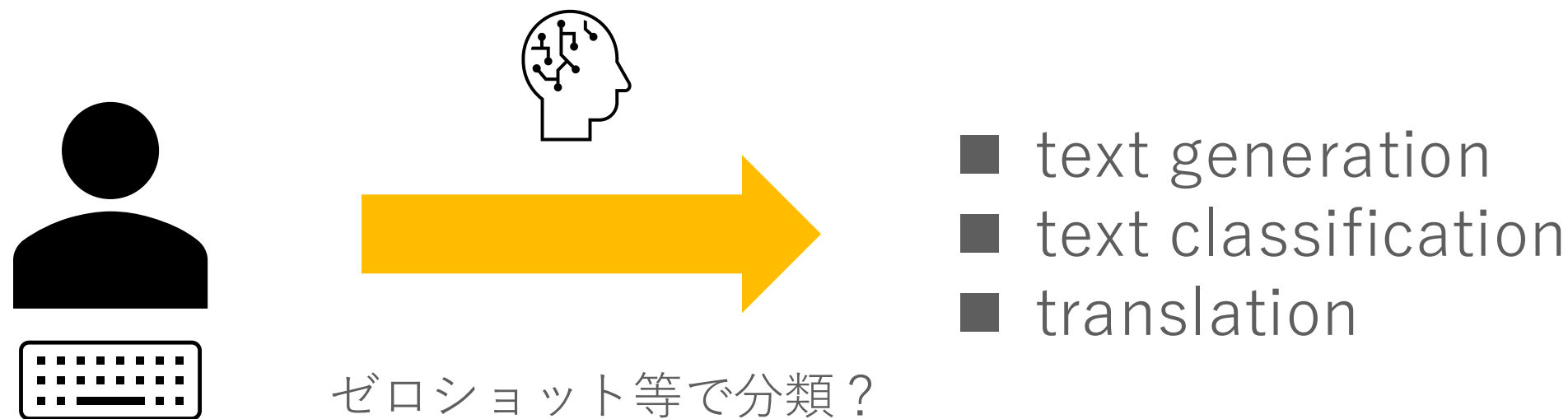
⇒自分でモデルの候補を選ばなくて良い.

現在考えているアプローチ

1. ユーザの質問文(クエリ)を取得
2. タスク分類
3. 推定タスクの中からダウンロード数上位のモデルを抽出
4. モデルのメタ情報を用い, モデルの候補を選定
5. (LLMとクエリの類似度をコサイン類似度を用いて計算する)

クエリを識別する

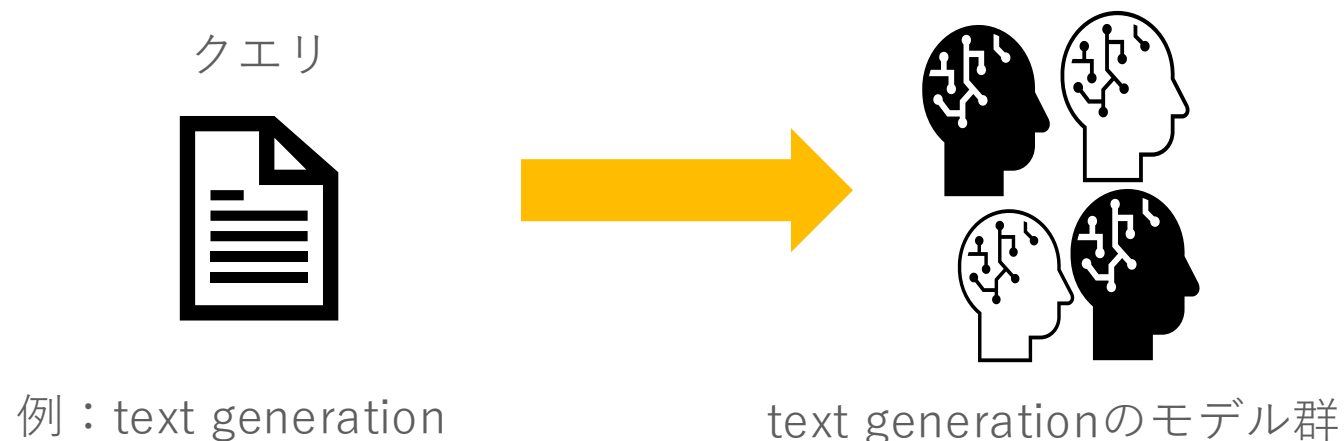
ユーザーが入力したクエリをタスク毎に分別



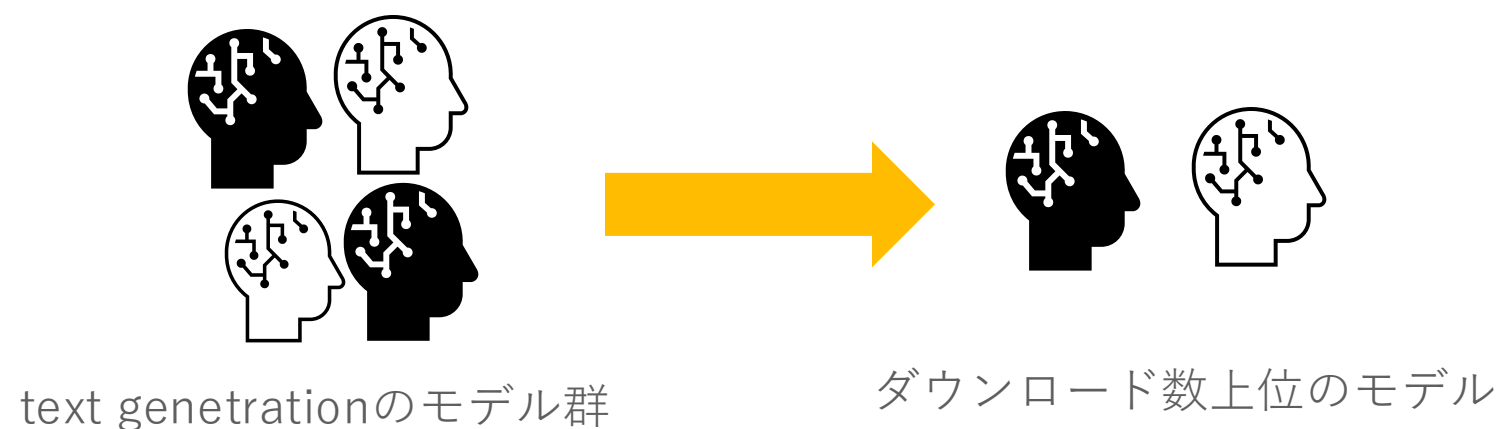
- Hugging faceで定義されているで定義されているNLPタスクに分類

対応するモデル群を抽出

タスクに対応するモデル群を取得

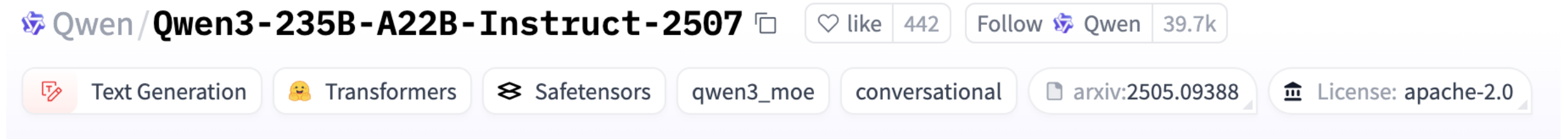


ダウンロード数が上位のモデルを抽出



モデル候補を選定

メタ情報(モデル名, タグetc...)を使用し,
モデル候補を選定



モデルの説明文等も考えたが、モデルの大半が
記載なし

Hugging face

AI技術のために提供されているオープンソースプラットフォーム

- 「Transformers」がベース
- AIモデルやデータを共有したり利用したりできる.
- LLMのGitHubのようなもの.

このプラットフォームの中からLLMを引っ張っていき たい

- どのような情報が入っているか

