

LLM学習用QAデータセットの 自動構築手法の提案

飯塚康太, 吉藤尚生 (株式会社フィックスターズ)

2024/03/15



本発表の概要

- カスタムLLM開発における課題
 - 学習したいデータは多種多様で、学習可能な形式に加工するのが大変
- 自動データセット構築手法の紹介
 - markdownファイルからLLMを利用して質問回答を生成するパイプラインを実装した
- データセット構築パイプラインの評価
 - モデル、プロンプト、温度パラメータによる出力の差について比較した
- 今後の展望

自己紹介

- 発表者：飯塚康太
 - 2019年に株式会社フィックスターズに新卒入社（今年で5年目）
 - 入社以来、車載チップ向けのモデル開発など深層学習を中心に取り組んできた
 - 機械学習コンテストサイト kaggle で Competitions Master の称号を獲得 ([@kotaiizuka](#))
 - 個人としては数学と情報科学の間の領域（証明支援系など）に興味あり
- 所属：株式会社フィックスターズ
 - Speed up your Business をコーポレートメッセージとし、ソフトウェア高速化事業を推進
 - LLMも「ビジネスの高速化」のために重要であると考えて [LLM R&D Center](#) を立ち上げ
 - 今回の発表内容を含むサービスを [Fixstars K4](#) として販売開始

カスタムLLM開発に おける課題

背景：データセットの変換が必要

- 既存の学習スクリプトは特定の形式のテキストデータを要求することが多い
 - LLMのファインチューニングによく使われる [alpaca dataset](#) や [llm-japanese-dataset](#) は質問回答が下記のような json フォーマットで記載されている。そのためデータをこの形式に変換すれば、既存のスクリプトでカスタムLLMが学習できるようになる

```
{  
  "instruction": "(質問)",  
  "input": "(多肢選択の場合は選択肢、自由回答の場合は空文字列)",  
  "output": "(回答)",  
}
```

llm-japanese-dataset のフォーマット

課題：データセットの変換に手間がかかる

- 一方で、学習させたいデータは通常このような形式になっていない
 - 学習させたいデータの例としては、次のような自然文が多いと考えられる

```
# sort
```

```
## 備考
```

- この関数には、特定のアルゴリズムで実装すべきという規定はない
- 実装のアルゴリズムとしては、クイックソートの改良版であるイントロソートが使われることが多い
- クイックソートは平均計算量が $O(N \log N)$ だが、最悪計算量が $O(n^{\sup}2^{\sup})$ である。そのため、C++03の計算量要件には合致するが、C++11の要件には合致しない

[cpprefjp](#) (C++日本語リファレンス) に含まれる文章の例 (markdown 形式)

- 単純なテキストデータ以外も学習させたいという要望もあるが、今回は対象外とする
 - PDF, word, html, jupyter notebook など
 - 表や図が含まれる場合

自動データセット構築 手法の紹介

自動データセット構築に必要な機能

- 入力のテキストを質問回答の形式に変換すること
 - 人力で作る場合も文章を精読する必要がある難しい部分で、LLMを使うのが適切
 - 実際 [alpaca dataset](#) は GPT-3 で生成した質問回答がもとになっている
- 質問回答の形式が機械可読なフォーマットになっていること
 - この部分については、モデルの出力に表記ゆれがあることが想定されるがルールベースの後処理で対応が可能なので、ここでは問題としない
 - 質問回答が箇条書きになっている、または json フォーマットになっている等後処理が簡単になるようなプロンプトを作ることは別途必要

入出力の例

- [cppref.jp](#) (C++日本語リファレンス) のドキュメントを変換した結果の一部

markdown text

- この関数には、特定のアルゴリズムで実装すべきという規定はない
- 実装のアルゴリズムとしては、クイックソートの改良版であるイントロソートが使われることが多い
- クイックソートは平均計算量が $O(N \log N)$ だが、最悪計算量が $O(n^2)$ である。そのため、C++03の計算量要件には合致するが、C++11の要件には合致しない

instruction

この情報を利用していくつか日本語で質問回答の例を作ってください。質問文を利用するときは上記のコンテキストが見れないので適切に主語を補ってください。フォーマットは下記の json形式としてください。

```
{"質問": "(質問文1)", "回答": "(回答文1)"}
```

```
{"質問": "(質問文2)", "回答": "(回答文2)"}
```



dataset

```
{  
  "質問": "std::sort 関数はどのように実装されていますか？",  
  "回答": "std::sort 関数は、クイックソートの改良版であるイントロソートが使われることが多いです。"  
  ...  
}
```

```
I hope these examples are helpful! Let me know if you have any other questions.
```

パイプラインを使ったときの作業効率化

- 手動で質問回答を作る場合
 - 今回のテキストに対して 6 個の質問回答を作るのに平均して約 5.5 分かかった (55秒/個)
 - 今回構築したパイプラインを使う場合
 - NVIDIA H100 80GB GPU を2台使用して並列処理を行った場合、4256個の質問回答を約10.4分で生成できた (0.15秒/個)
- 自動化することで約380倍の高速化ができた
- パイプラインの実行中は放置できるので、他の作業が進められるのも利点

自動データセット構築 パイプラインの評価

パイプラインの評価実験

- 「入力のテキストを質問回答の形式に変換すること」という要件をパイプラインの出力が満たしているか確認した
 - 出力に対して2つの観点で評価し、5段階で採点した
 - 質問回答の形式になっているか？
 - 情報の誤りや過不足がないか？

	すべて正しい	誤りが1個ある	誤りが複数ある
質問回答形式である	5点	4点	3点
質問回答形式でない	2点	1点	

今回の実験における採点基準

パイプラインの評価実験

- 重みが公開されているLLMは多数あり、どれがQA生成に適切か調査が必要
- モデル選定のために各種リーダーボードで上位のモデルを確認した
 - [AlpacaEval 2.0](#) : 複数のモデルの出力を GPT-4 Turbo に比較させた結果
 - [Harness](#) : 常識的な質問や計算問題などに対する回答精度の平均
- その中からパラメータ数ごとに下記の5種類のモデルを選択した
 - zephyr7b (7B) : [HuggingFaceH4/zephyr-7b-beta](#)
 - xwin13b (13B) : [Xwin-LM/Xwin-LM-13B-V0.2](#)
 - yi34b (34B) : [01-ai/Yi-34B-Chat](#)
 - mixtral8x7b (46B) : [mistralai/Mixtral-8x7B-Instruct-v0.1](#)
 - beluga70b (70B) : [stabilityai/StableBeluga2](#)
 - 本研究での評価結果もパラメータ数やリーダーボードと正の相関があると予想される

パイプラインの評価実験

- プロンプトは8種類作成した
 - <https://github.com/lm-sys/FastChat/> のデフォルトプロンプトを応用したもの
 - fastchat, fastchat+json, fastchat+json+jp, normal_json
 - <https://www.promptingguide.ai/jp> の内容を応用したもの
 - with_FAQ, teacher-student_json, abstract, simple
 - モデルの種類によって適切なプロンプトが異なる可能性もあり、どれが良いプロンプトか一概には決まらなると予想される
- 温度パラメータを 0.2, 0.4, 0.6, 0.8, 1.0 の5種類で評価した
 - 高いほど創造的な回答をし、低いほど固定された回答となるので、正確性が重要な今回の用途では低いほうが精度が上がると予想される
- 各パラメータで1回ずつ、計200個の出力について採点した

パイプラインの評価例

モデル: xwin13b

プロンプト: simple

温度: 0.6

- 質問回答の形式になっている (3~5点)
- 計算量要件の説明で、主語がクイックソートからイントロソートに変わってしまった (誤り)
- 戻り値に対する質問回答が抜けている (誤り)

→ 誤りが複数あるので 3点 とする

1. このドキュメントは、C++のsortアルゴリズムについて説明しています。

質問: 1. sortアルゴリズムは、どのようなイテレータをサポートしますか?

回答: sortアルゴリズムは、RandomAccessIteratorをサポートします。このクラスは、ValueSwappableの要件を満たしている必要があります。また、firstの型は、MoveConstructibleとMoveAssignableの要件を満たしている必要があります。

1. このアルゴリズムは、どのような範囲をソートすることができますか?

回答: sortアルゴリズムは、[first, last)の範囲をソートすることができます。ここで、firstとlastは、イテレータである必要があります。

1. この関数の計算量はどれくらいですか?

回答: C++03では、平均して約 $N \log N$ ($N == \text{last} - \text{first}$) 回の比較です。C++11以降では、 $O(N \log N)$ ($N == \text{last} - \text{first}$) 回の比較です。ただし、この関数には特定のアルゴリズムで実装すべきという規定はなく、実装のアルゴリズムとしては、イントロソートがよく使われます。イントロソートは、平均計算量が $O(N \log N)$ だが、最悪計算量が $O(n^2)$ であるため、C++03の計算量要件には合致するが、C++11の要件には合致しないことに注意してください。

1. このアルゴリズムの実装で使われるアルゴリズムは、どのようなものですか?

回答: 実装のアルゴリズムとしては、クイックソートの改良版であるイントロソートがよく使われます。イントロソートは、平均計算量が $(N \log N)$ であり、最悪計算量も $O(N \log N)$ です。しかし、C++03では最悪計算量が $O(n^2)$ であるため、C++11以降の要件を満たしていません。

パイプラインの評価例

モデル: yi34b

プロンプト: fastchat+json+jp

温度: 1.0

- json形式だが質問回答の形式にはなっていない (1~2点)

→ 内容は過不足なく正しいので 2点 とする
(言語の差は今回の判定基準に含まれない)

```
```json
{
 "algorithm": "std::sort",
 "summary": "Sort the range [first, last)";
 "requirements": "RandomAccessIterator and ValueSwappable
for *first, MoveConstructible and MoveAssignable for the type
of *first",
 "effect": "Sort the range [first, last)";
 "return": "None",
 "complexity": "C++03: Average case N log N comparisons,
C++11 and later: O(N log N) comparisons";
 "remark": "No specific algorithm to implement this
function. Introsort, an improvement of quicksort, is often
used."
}
```
```

実験結果：モデルによる影響

- `mixtral8x7b` が全体として高い性能を示した
 - 5点満点の回答が特に多く、適切な後処理を自動で行ったのち学習に利用できると考えられる

| モデル名 | 1点 | 2点 | 3点 | 4点 | 5点 | 平均点 |
|-------------|----|----|----|----|----|------|
| zephyr7b | 6 | 7 | 15 | 11 | 1 | 2.85 |
| xwin13b | 0 | 2 | 28 | 10 | 0 | 3.20 |
| yi34b | 1 | 5 | 14 | 17 | 3 | 3.40 |
| mixtral8x7b | 0 | 4 | 12 | 10 | 14 | 3.85 |
| beluga70b | 5 | 3 | 16 | 13 | 3 | 3.15 |

プロンプト8種類・温度パラメータ5種類に対するモデルごとの点数分布

実験結果：既存指標との相関

- Harness の評価 は点数との相関が比較的強くみられた
 - Harness は質問に対して正しく回答する性能を見るタスクが多いため、質問生成と近い評価ができると推測できる

| モデル名 | サイズ | AlpacaEval 2.0 | Harness | 本研究での平均点 |
|-------------|------------|----------------|--------------|-------------|
| zephyr7b | 7B | 10.99% | 59.08 | 2.85 |
| xwin13b | 13B | 17.43% | 記載なし | 3.20 |
| yi34b | 34B | 29.66% | 65.32 | 3.40 |
| mixtral8x7b | 46B | 18.26% | 68.47 | 3.85 |
| beluga70b | 70B | 記載なし | 67.42 | 3.15 |

モデルサイズ・リーダーボード（AlpacaEval 2.0, Harness）と本研究での平均点

実験結果：プロンプトによる影響

- few-shot prompting を使ったものは点数が低い傾向
 - タスクにあった入出力例を適切に配置する必要があるため難易度が高い

| プロンプト名 | システムプロンプトの文字数 | 指示文の文字数 | 平均点 |
|----------------------|---------------|---------|------|
| fastchat | *2246 | 94 | 3.16 |
| fastchat+json | *2246 | 117 | 3.28 |
| fastchat+json+jpg | *2246 | 128 | 3.04 |
| with_FAQ | 66 | *859 | 2.64 |
| normal_json | 61 | 207 | 3.80 |
| teacher-student_json | 42 | 276 | 3.48 |
| abstract | 66 | 171 | 3.28 |
| simple | 66 | 85 | 3.64 |

プロンプトの文字数と平均点 *few-shot prompting（プロンプトに質問回答の類似例を与える手法）を使用

パイプラインの評価例

- 具体的すぎる指示を減らすと精度が上がる傾向

あなたは役に立つ人工知能アシスタントです。ユーザの質問、依頼を正確に答えてください。

ユーザー：これは、C++ について書かれた文書の一部です：

`{{page}}`

上記の文章について日本語での質問文と返答文のセットを作り、```"conversations": [{"生徒": "<質問1>", "先生": "<回答1>"}, {"生徒": "<質問2>", "先生": "<回答2>"}, {"生徒": "<質問3>", "先生": "<回答3>"}, {"生徒": "<質問4>", "先生": "<回答4>"}]```のように4つ以上の質問と答えを考え、それをJSON形式で返しなさい。ダブルクォーテーションは適切にエスケープしなさい

アシスタント：

teacher-student_json prompt (3.48点)

好奇心旺盛な人間と人工知能アシスタントとの会話。アシスタントは人間の質問に対して、役立つ、詳細かつ丁寧な答えを返します。

Human: コンテキストを提示します

`{{page}}`

この情報を利用していくつか日本語で質問回答の例を作ってください。質問文を利用するときは上記のコンテキストが見れないので適切に主語を補ってください。フォーマットは下記のjson形式としてください。

```
{"質問": "(質問文1)", "回答": "(回答文1)"}
```

```
{"質問": "(質問文2)", "回答": "(回答文2)"}
```

Assistant:

normal_json prompt (3.80点)

パイプラインの評価例

- 具体的すぎる指示を減らすと精度が上がる傾向

好奇心旺盛なユーザーと人工知能アシスタントとのチャット。アシスタントは、ユーザーの質問に対して、親切かつ詳細かつ丁寧に回答します。

ユーザー：これは、C++ について書かれた文書の一部です：

`{{page}}`

では、このドキュメントについて質問と回答の組を書いてください。

ただし、質問は「関数Xの機能は何ですか？」や「関数Yを使って何ができますか？」などの形式を使ってください。

回答は正確であることに努め、できればコードで例示してください。

アシスタント：

abstract prompt (3.28点)

好奇心旺盛なユーザーと人工知能アシスタントとのチャット。アシスタントは、ユーザーの質問に対して、親切かつ詳細かつ丁寧に回答します。

ユーザー：これは、C++ について書かれた文書の一部です：

`{{page}}`

では、このドキュメントについて質問と回答の組を書いてください。

アシスタント：

simple prompt (3.64点)

実験結果：モデルとプロンプトの相関

- モデルを変えると最大3.0点も性能が変わるプロンプトがある
 - モデルを変えたらプロンプトも一緒に変えて比較することが重要

| | zephyr7b | xwin13b | yi34b | mixtral8x7b | beluga70b | 最大-最小点 |
|----------------------|----------|---------|-------|-------------|-----------|--------|
| fastchat | 1.6 | 3 | 3.4 | 4.6 | 3.2 | 3.0 |
| fastchat+json | 2.4 | 3.6 | 3.2 | 4.4 | 2.8 | 2.0 |
| fastchat+json+jpg | 3 | 3.2 | 3.2 | 2.8 | 3 | 0.4 |
| normal_json | 3.6 | 3.4 | 3.8 | 4.8 | 3.4 | 1.4 |
| with_FAQ | 2.2 | 2.6 | 3 | 2.8 | 2.6 | 0.8 |
| teacher-student_json | 3.4 | 3.4 | 3.8 | 3.4 | 3.4 | 0.4 |
| abstract | 3.6 | 3.2 | 3.2 | 3.8 | 2.6 | 1.2 |
| simple | 3 | 3.2 | 3.6 | 4.2 | 4.2 | 1.2 |
| 平均点 | 2.85 | 3.2 | 3.4 | 3.85 | 3.15 | 1.0 |

モデルとプロンプトごとの平均点

実験結果：温度パラメータによる影響

- モデルによっては、温度を変えた場合も最大1.1点の差がある
 - 仮説通り温度は低いほうが良い傾向がある
 - プロンプトと同じく温度も、モデルを変更したらパラメータ探索が必要

| | 0.2 | 0.4 | 0.6 | 0.8 | 1.0 | 最大-最小点 |
|-------------|-----|-----|-----|-----|-----|--------|
| beluga70b | 3.6 | 3.4 | 3.1 | 2.5 | 3.1 | 1.1 |
| mixtral8x7b | 3.6 | 3.9 | 4.4 | 3.6 | 3.8 | 0.6 |
| xwin13b | 3.3 | 3.5 | 3.3 | 3.0 | 3.0 | 0.5 |
| yi34b | 3.8 | 3.6 | 3.6 | 3.0 | 3.0 | 0.8 |
| zephyr7b | 2.9 | 3.3 | 2.5 | 3.1 | 2.5 | 0.8 |
| 平均点 | 3.4 | 3.5 | 3.4 | 3.1 | 3.1 | 0.4 |

モデルと温度パラメータごとの平均点

実験結果：まとめ

- テキストから質問回答を自動で作るパイプラインを構築した
 - 手動で質問回答を作るのに比べて380倍程度速い
- モデル・プロンプト・温度パラメータの3項目を変えてパイプラインを実行し、質問回答の正確性という観点で5段階評価を実施した
- 3項目すべてで、生成される質問回答の質が大きく変化するのを確認した
 - 今回の実験では、モデルは `mixtral8x7b`, プロンプトは `normal_json`, 温度パラメータは `0.4` に設定するともっとも得点の高い出力が得られ、データセットとして使用可能な高品質の出力であることを確認できた
 - 今回の実験では、データセットやモデルパラメータの多くを固定したが、これらによっても傾向が変化することが想定される

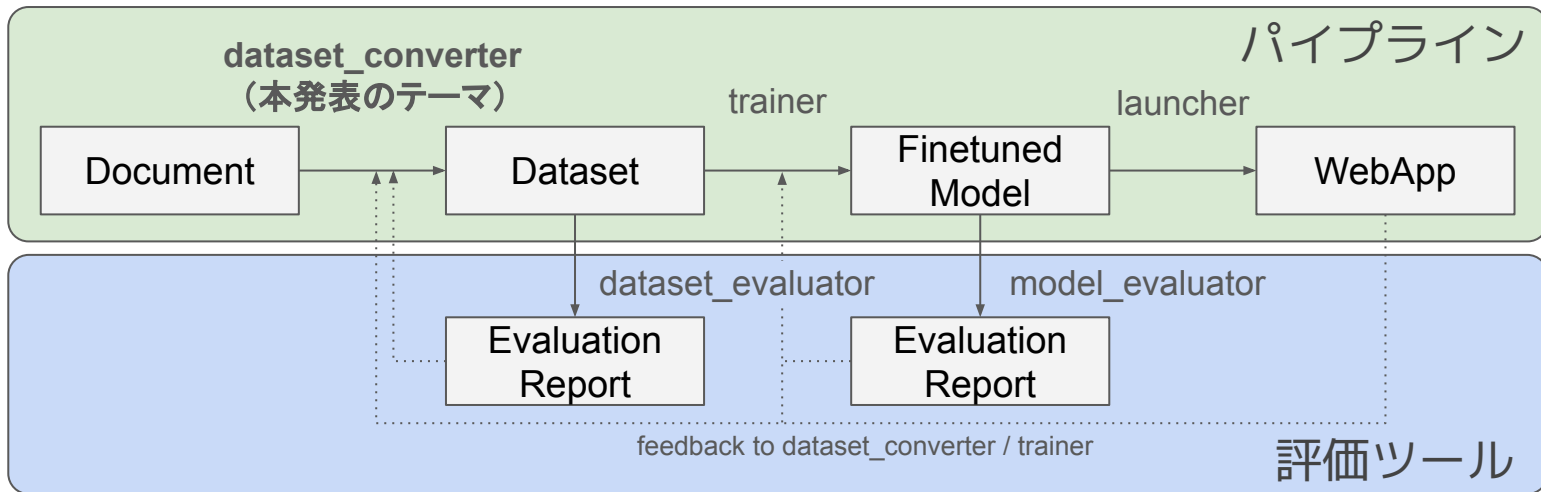
課題

- 質問回答データセットの生成は自動でできるようになった一方で、データセットやモデルの出力を手動評価するのが大変
 - 今回は200通りの出力に対して目視で確認し点数付けするのに4時間程度かかった
 - より大きなデータセットの手動評価は長時間かかる
 - 本来は各データセットで学習したモデルの応答を評価したいが、手間がかかる
 - データセットの正確性や学習したモデルの応答を自動で評価するシステム作りが必要

今後の展望

今後の展望

- LLMを利用したデータセット・モデルの自動評価パイプラインを実装中
 - データセットに含まれる質問回答やモデルの応答の正確性を検証する機能を追加



- 精度や速度に課題があり、改善を今後していく予定

謝辞：本研究では、さくらインターネットとの共同研究開発において
お借りしているサーバーを利用させていただきました