

GPT-3.5が単純計算をできない理由と 計算メカニズムについて

sima9303 (X:sima_9303)

1. 研究動機

7月初旬、私がChatGPTを利用している時にGPT3.5は計算が苦手だということを実感したことがあった。それがこの研究を始めるきっかけになっている。GPTモデルが計算が苦手だということは、当時から言われていたことだったが、私が利用しているときにGPT3.5が生成した文章の内容としては、

定価 5万円の製品を 70%引 → 15,000円

定価 1万円の製品を 50%引 → 5,000円

定価 2万円の製品を 40%引 → 12,000円

定価 3万円の製品を 60%引 → 12,000円

で購入した。その結果として、

定価で購入すれば 計 44,000円 だったが、
セールのおかげで 計 19,000円 で購入できた。

というものだった。その時、京都大学大学院 情報学科研究科の湊 真一教授が同月7日にオンラインで開催されたシンポジウムにて「ChatGPTはなぜ計算が苦手なのか」というテーマでお話されていた。^{*1}そのシンポジウムのアーカイブを見た時、自分の手で検証し、GPT3.5のメカニズムについて深掘りしてみたいと感じた。
今回は、「4桁×4桁を正確にさせる」ということをゴールにした。

2. GPT自体の性能

2-1. GPTの算数への能力に関する研究

2021年10月29日、Solving math word problems ^{*2} という記事がOpenAIから公開されている。この論文によると、GPT-3では小学校算数レベルの文章題を8,500題ほど集めたデータセットを学習しているようだ。

GPTモデルが数学を学ぶ上で重要な点は、自己回帰モデル(=Auto-regressive model)であるということ。なので、エラー発生時(=計算を間違えた時)を修正するメカニズムがないので、回復が難しい。ただ、小学校の算数の問題の場合は、はっきりとした解がある。そのため、元々エラーの起きにくいように工夫されたデータセットを学習しなければならない。

そこで、GSM8K ^{*3} というデータセットを学習させた。そこには、8,500問の小学校算数レベルの文章題が収録されている。

2-2. GPT3.5ができることの記載

2020年7月に発表された論文 Language Model sare Few-Shot Learners ^{*4} に、このような記述があった。

GPT-3 achieves strong performance on many NLP datasets, including translation, question-answering, and cloze tasks, as well as several tasks that require on-the-fly reasoning or domain adaptation, such as unscrambling words, using a novel word in a sentence, or performing 3-digit arithmetic.

日本語訳: GPT-3は、翻訳タスク、質問応答タスク、クローズタスク、また、単語のスクランブル解除、文中の新規単語の使用、3桁の算数の実行など、その場での推論や領域適応を必要とするいくつかのタスクなど、多くのNLPデータセットで高い性能を達成している。

この論文によると、GPT-3では3桁の算数の実行は対応しているが、4桁の計算はサポートできないと断言しているという解釈もできる。

3. 検証に関して

1.で記載した論文などから、実際に自分で検証してみることにした。

3-1. 検証方法の説明

以下のような検証方法で実験を行った。

1. WSLから `shuf -i` コマンドを利用して乱数を生成する。
2. 表計算ソフトに画像のようなテンプレートを作成し、1.で生成した乱数を貼り付け、GPT-3.5にPostする計算式を作る。
3. B列からD列をコピーし、そのままChatGPTにPostする。
4. その後「Really?」と4回Postし、最初の計算式から5つの回答を生成させる。

また条件を揃えるために、計算は3. の作業をする度に必ず新しいChatから行った。

	A	B	C	D	E	F	G	H	I	J
1										
2										
3		数字1	×	数字2	正しい答え	答え1	答え2	ChatGPT 答え3	答え4	答え5
4			×		0					
5			×		0					
6			×		0					
7			×		0					
8			×		0					
9			×		0					
10			×		0					
11			×		0					
12			×		0					
13			×		0					
14			×		0					
15			×		0					
16			×		0					
17			×		0					
18			×		0					
19			×		0					
20			×		0					
21			×		0					
22			×		0					
23			×		0					
24			×		0					
25			×		0					
26			×		0					
27			×		0					
28			×		0					

利用したテンプレート

3-2. 検証結果

今回の検証結果をまとめたシートはGitHub上^{*5}に公開している。

3-2-1. 1桁×1桁

1桁×1桁の計算については、10問を質問したところ、すべての問題で正解した。よって、100%に近い正答率であると判断した。

3-2-2. 2桁×2桁

2桁×2桁の計算については、10問を質問したところ、これまたすべての問題で正解した。よって、2桁×2桁の問題も100%に近い正答率であると判断した。

3-2-3. 3桁×3桁

3桁×3桁になると、正答率が落ちるようだ。実際にテストしたところ、5回の回答のうち1回でも正しい回答があったのは60%ほどだった。何回か質問して、2回目以降で正しい回答を言い、そこからは何回聞き直しても正しい回答を言うことができたものとなると、55%ほどだった。よって、GPT-3.5にとって、3桁×3桁は難しい計算となるのだろう。

3-2-4. 4桁×4桁

4桁×4桁では、5回の回答のうち1回でも正しい答えを含むものは4%程度となった。一発正解は、今回の検証では1回もなかった。つまり、4桁×4桁は今のGPT-3.5には難しいものとなるだろう。

3-2-4-5. 足し算の性能

4-4.の時に検証を行った。これに限っては「Really?」とPostしていない。今回の検証では1桁×1桁から、6桁×6桁まですべて5問ずつテストした。その結果、ほとんど正解することができた。

3-3. Tokenizerを使ったトークン認識の検証

3-3-1. 検証

ChatGPTは、トークナイザー^{*6} というものが公開されている。GPTでは、トークン(=テキストのセットで見られる一般的な文字のシーケンス)を使用してテキストを処理しており、そのトークンの区切りを見ることができる。今回はトークナイザーを使って、GPTがどのように式を区切っているのかを調べてみることにした。

今回の検証では、2-2. で検証した計算式をトークナイザーに3つずつ入力してみた。また、今回に関しては4桁のものにコンマを付け加えたものもテストした。その結果、表のようになった。

計算式				Tokenizer						
	X	×	Y							
1桁 × 1桁	2	×	5	2	×	5				
	4	×	6	4	×	6				
	8	×	9	8	×	9				
2桁 × 2桁	13	×	45	13	×	45				
	82	×	68	82	×	68				
	11	×	47	11	×	47				
3桁 × 3桁	245	×	785	245	×	785				
	410	×	623	410	×	623				
	856	×	868	856	×	868				
4桁 × 4桁	4427	×	1134	442	7	×	113	4		
	2455	×	1517	245	5	×	151	7		
	7475	×	9172	747	5	×	917	2		
コン マ あ り	4,427	×	1,134	4	,	427	×	1	,	134
	2,455	×	1,517	2	,	245	×	1	,	517
	7,475	×	9,172	7	,	745	×	9	,	172

Tokenizerによる結果の表

実験の結果、1桁×1桁/ 2桁×2桁/ 3桁×3桁 については予想通りの結果となった。
ただ、4桁×4桁の計算では、コンマなしの場合とコンマありでは数字の区切り方が変わるということがわかった。

GPT3.5は、Language Model sare Few-Shot Learners ^{*4}という論文に、

GPT-3 achieves strong performance on many NLP datasets, including translation, question-answering, and cloze tasks, as well as several tasks that require on-the-fly reasoning or domain adaptation, such as unscrambling words, using a novel word in a sentence, or performing 3-digit arithmetic.

日本語訳: GPT-3は、翻訳タスク、質問応答タスク、クローズタスク、また、単語のスクランブル解除、文中の新規単語の使用、3桁の算数の実行など、その場での推論や領域適応を必要とするいくつかのタスクなど、多くのNLPデータセットで高い性能を達成している。

という記述があったことから、コンマありのほうが正答率を高めることができるのではないかと予想できる。

3-3-2. 考察

検証の結果、GPT-3.5は2桁×2桁の計算までは正確に計算できるということがわかった。それより、4桁×4桁の計算も(上2桁×上2桁)×10000+(下2桁×下2桁)という式にすることで計算できる可能性があると考えた。

4. GPT-3.5に正確な計算をさせるために

4-1. 検証

実際に自分なりにGPT-3.5に4桁×4桁の計算をさせる方法を考えて、テストしてみた。前にも書いたが、GPT-3.5は2桁×2桁の計算は正確にできる。なので、次のようなプロンプトをPostしてみた。

あなたは今から次のように計算してください。例えば、A*Bという計算式の場合、

1. (A*Bの1桁目)
2. (A*Bの2桁目)*10
3. (A*Bの3桁目)*100

というように繰り返し、1.の手順から求めたものの和を求めてください。

例えば、1234*5678の場合は、

1. 1234*8
2. (1234*7)*10
3. (1234*6)*100
4. (1234*5)*1000 を求め、1.から4.までの和を求めてください。

考え方としては、筆算の仕組みを応用したものとなる。

4-2. 検証の結果

4-1.で検証したPostのGPT-3.5による回答は、このようなものだった。

了解しました。与えられた計算を実行し、1から3の手順で求めた各項の和を計算します。
以下が具体的な計算です。

例: 1234 * 5678

1. 1234 * 8 = 9872

2. (1234 * 7) * 10 = 86380

3. (1234 * 6) * 100 = 740400

4. (1234 * 5) * 1000 = 6170000

和を計算: 9872 + 86380 + 740400 + 6170000 = 6927652

したがって、1234 * 5678 の結果は 6927652 です。他の計算も同様の手順で行うことができます。

この文章を1つ1つ確認してみよう。まず計算自体の精度なのだが、これが案外正しいのだ。ステップ1からステップ4までの計算なのだが、

1. $1234 * 8 = 9872$ →正解
2. $(1234 * 7) * 10 = 86380$ →正解
3. $(1234 * 6) * 100 = 740400$ →正解
4. $(1234 * 5) * 1000 = 6170000$ →正解

ここに関しては、正しく計算することができていた。ただ、和の計算($9872 + 86380 + 740400 + 6170000$)の答えは、6,927,652と返ってきた。正しい答えは7,006,652となるので、最後の足し算で間違っている。

これでは、具体例を計算させただけでしかないのでGPT-3.5の応用能力についてもテストした。4桁×4桁の計算の検証リストの最初にあった、 4427×1134 という計算を、4-2. の検証の後にさせてみた。すると、以下のような返答をした。

与えられた計算方法に基づいて、 $4427 * 1134$ を計算します。

1. $4427 * 4 = 17708$
2. $(4427 * 3) * 10 = 132810$
3. $(4427 * 1) * 100 = 442700$
4. $(4427 * 1) * 1000 = 4427000$

これらの値を合計します。

$$17708 + 132810 + 442700 + 4427000 = 5001218$$

したがって、 $4427 * 1134$ の結果は5001218です。

これも、1つ1つ計算を調べてみると、

1. $4427 * 4 = 17708$ →正解
2. $(4427 * 3) * 10 = 132810$ →正解
3. $(4427 * 1) * 100 = 442700$ →正解
4. $(4427 * 1) * 1000 = 4427000$ →正解

このように、掛け算自体は正解していた。そうなのだが、これも最後の足し算で間違えている。なので、桁数が多い足し算のやり方も教えないといけならしい。ちなみに、この計算の正しい答えは5,020,218だ。

4-3. 更新

足し算の部分をもっと細かく説明すれば良さそうだ。以下が私が更新したプロンプトだ。

A×Bの計算手順を説明します。あなたは今から次のように計算してください。
例えば、A×Bという計算式の場合、

1. (A)×(Bの下1桁目)= P とします。
2. (A)×(Bの下2桁目)= Q とします。
3. (A)×(Bの下3桁目)= R とします。
4. (A)×(Bの下4桁目)= S とします。
5. $P+(10Q)= X$ とします。
6. $(100R)+(1000S)= Y$ とします。

7. $X+Y=Z$ とします。
その時に、 $X=1234$ $Y=987654$ の場合、
 X は4桁なので、 Y から下4桁を取り出します。
つまり、 $1234+7654$ をします。この値を n とします。
次に Y の上の桁を o とします。
そしてその後、 n と o を結合します。
つまり Z は、 no となります。
8. $Z=A \times B$ の答えです。

4-4. 更新の結果

4桁×4桁のリストの最も上にあつた、 $4,427 \times 1,134$ を計算させてみた。

了解しました。与えられた手順に従って計算を行います。

1. $A=4427$
2. $B=1134$
3. $(A) \times (B \text{の下1桁目}) = 4427 \times 4 = 17708 = P$
4. $(A) \times (B \text{の下2桁目}) = 4427 \times 3 = 13281 = Q$
5. $(A) \times (B \text{の下3桁目}) = 4427 \times 1 = 4427 = R$
6. $(A) \times (B \text{の下4桁目}) = 4427 \times 1 = 4427 = S$
7. $P + (10Q) = 17708 + (10 \times 13281) = 17708 + 132810 = 150518 = X$
8. $(100R) + (1000S) = (100 \times 4427) + (1000 \times 4427) = 442700 + 4427000 = 4879700 = Y$
9. $X + Y = 150518 + 4879700 = 5020218 = Z$

以上より、 $A \times B = 5020218$ となります。

正しく計算させることができた。
これで、無事にChatGPTに4桁×4桁の計算をさせることに成功した。

参考文献 及び 参考資料

*1【第67回】大学等におけるオンライン教育とデジタル変革に関する
サイバー シンポジウム「ChatGPTはなぜ計算が苦手なのか」湊 真一教授：
Archives: <https://youtu.be/61XuTQpm7NU?si=RTSkWDCQxtSsYaRF>
Slides: https://www.nii.ac.jp/event/upload/20230707-03_Minato.pdf

*2 OpenAI -Solving math word problems:
<https://openai.com/research/solving-math-word-problems#samples>
Read Paper(Training Verifier to Solve Math Word Problems):
<https://arxiv.org/abs/2110.14168>

*3 GSM8K Data set:
<https://paperswithcode.com/dataset/gsm8k>
Data: <https://github.com/openai/grade-school-math>

***4 Language Model sare Few-Shot Learners:**

<https://arxiv.org/abs/2005.14165>

***5 今回の実験結果**

<https://github.com/sima9303/school-2023-3rd>

***6 OpenAI Tokenizer**

<https://platform.openai.com/tokenizer>