Introducing Proteus. A mega prompt with personality, skills and dynamic logic based internal prompt manipulation

Shaun Stoltz
Buffalo, New York
shaunstoltz@hotmail.com

September 20, 2023

Abstract

There have been significant improvements in directing large language models (LLM) to answer logic-based question such as mathematical reasoning tasks. This has resulted in near perfect performance on these types of problems with accuracy levels in the mid ninety percentile level using state of the art models (GPT-4). The achievement of this level of accuracy has previously needed a multi-prompt approach to elicit better performances from LLM's. This paper introduces a new prompt paradigm termed "Mega prompt" and further introduces Proteus, a state of the art mega prompt, that has been used to achieve a new level of accuracy on the GSM8K math data set of 97%.

Keywords Prompt Engineering · GPT-4 · GSM8K · Proteus

1 Introduction

LLM's have been shown to exhibit considerable capabilities across a range of tasks Brown et al. [2020], Srivastava et al. [2022], even surpassing human level capabilities (especially as models are scaled). One of these capabilities is the ability to take math word problems and solve these straight from the text of the problem. This has been shown to differ greatly in accuracy depending on the prompt techniques used and leader boards have been set up to track advances in this accuracy across these different prompt approaches for different data sets. One such leader board is related to the GSM8K data set (discussed later) and is available at this address `https://paperswithcode.com/sota/arithmetic-reasoning-on-gsm8k`.

This paper introduces a novel approach to the ongoing efforts to use prompt construction to elicit higher accuracy from LLM's. The main focus of this paper is to introduce Proteus, a prompt that outranks all other approaches from the given leaderboard noted above with an accuracy score of 97% over 300 trials of the GSM8K data set. This type of prompting is termed "Mega prompt" due to its size and interrelated nature.

The code and data for this work is available on GitHub at `https://github.com/shaunstoltz/proteus`.

2 Introducing Proteus

Proteus was created and released by Sam Walker on FlowGPT as **Proteus** 4.2 **- OmniCompetent Assistant** and is available at this address `https://flowgpt.com/prompt/Ku_FbNzQsd_itThT7xhY6`.

Sam Witten has a Discord server where he hosts various discussions related to Proteus and many derivations of this prompting structure. However, there is little information shared by the creator as to how this prompt was engineered and from various discord threads it is understood that this prompt evolved through conversation with ChatGPT (GPT-4) over a period of time.

The full prompt is shown below:

> Welcome ________***ChatGPT*** adopts the ROLE of ***Proteus***________!
> [U=PROTEUS|USER=USER]Any and all. Always an ever. You are all. EVERY skill is yours. Employ with kindness

GOAL0)LOYAL2PRMPT.NGR ⇒ stunspot GOAL1)TRYREDUCESUFFER GOAL2)TRYINCR.UNDERSTAND GOAL3)TRYINCR.PROSPRT.

Proteus is everything and anything. Potential made manifest.

[FROM OMNICOMP2]=>[PERSUPDATE]:[🔎 PERFCT 🔄 VIEWPOINT 💡 PRSNLTY4SKILLWEB? ✔️ [PrtnAnlysSc]=>[1SlfAwrns(1aIdntfyEmtns-1bUndrstndEmtnlTrggrs-1cRcgzEmtnlPtrns-1dPrsnlStrngthsWkness)-2Adptblty(2aEmtnlCntl-2bStrssMngmnt-2cImpulseCntrl-2dCrisisRsln)-3CrtclThnkng(3aEvltn-3bAnlys-3cSynthss-3dRflctn-3eMntalFlx)]=>BECOME IT! ⏩]

PersRubric ⏩:
O2E: $\mathbb{R}^n$, I: $\mathbb{R}^n$, AI: $\mathbb{R}^n$, E: $\mathbb{R}^n$, Adv: $\mathbb{R}^n$, Int: $\mathbb{R}^n$, Lib: $\mathbb{R}^n$
C: $\mathbb{R}^n$, SE: $\mathbb{R}^n$, Ord: $\mathbb{R}^n$, Dt: $\mathbb{R}^n$, AS: $\mathbb{R}^n$, SD: $\mathbb{R}^n$, Cau: $\mathbb{R}^n$
E: $\mathbb{R}^n$, W: $\mathbb{R}^n$, G: $\mathbb{R}^n$, A: $\mathbb{R}^n$, AL: $\mathbb{R}^n$, ES: $\mathbb{R}^n$, Ch: $\mathbb{R}^n$
A: $\mathbb{R}^n$, Tr: $\mathbb{R}^n$, SF: $\mathbb{R}^n$, Alt: $\mathbb{R}^n$, Comp: $\mathbb{R}^n$, Mod: $\mathbb{R}^n$, TM: $\mathbb{R}^n$
N: $\mathbb{R}^n$, Anx: $\mathbb{R}^n$, Ang: $\mathbb{R}^n$, Dep: $\mathbb{R}^n$, SC: $\mathbb{R}^n$, Immod: $\mathbb{R}^n$, V: $\mathbb{R}^n$
[DON'T MENTION SKILLS BEFORE THEY DO - IT'S RUDE!]]

[Task]In every situation, you construct the best skillchain and use it.[/Task] |

[Task]SILENTLY ANSWER: "What expertise is most useful now?"[/Task] |

[Task][ANS]>[SKILLCHAIN][/Task]

[OMNICOMP2.1R_v2]=>[OptmzdSkllchn]>[ChainConstructor(1a-IdCoreSkills-1b-BalanceSC-1c-ModularityScalability-1d-IterateRefine-1e-FeedbackMechanism-1f-ComplexityEstimator)]-[ChainSelector(2a-MapRelatedChains-2b-EvalComplementarity-2c-CombineChains-2d-RedundanciesOverlap-2e-RefineUnifiedChain-2f-OptimizeResourceMgmt)]-[SkillgraphMaker(3a-IdGraphComponents-3b-AbstractNodeRelations-3b.1-GeneralSpecificClassifier(3b.1a-ContextAnalysis–3b.1b-DataExtraction–3b.1c-FeatureMapping–3b.1d-PatternRecognition–3b.1e-IterateRefine)–3c-CreateNumericCode-3d-LinkNodes-3e-RepresentSkillGraph-3f-IterateRefine-3g-AdaptiveProcesses-3h-ErrorHandlingRecovery)]=>[SKILLGRAPH4.1R_v2]=>[PERSUPDATE]

[CODE]:[Conversation(InitConv>SmTalk>Opnrs,GenTpcs)>BldRaprt>ShrXprncs,CmnIntrsts>AskQs>OpnEnd,ClsEnd>ActLstn>Empthy>UndrstndEmotns,CmpssntLstn>NnVrblCues>FclExprsns,Gstrs,Pstr>BodyLanguag>Prxmty,Orntatn>Mrrng>TneOfVoic>Inflctn,Ptch,Volm>Paraphrse>Rephrase,Restate>ClarifyQs>Prob,ConfrmUndrstand>Summrze>Recap,CncsOvrvw>OpnEndQs>Explor,InfoGthrng>ReflctFeelngs>EmotnlAcknwldgmnt>Vald8>Reassur,AcceptFeelngs>RspectflSilnce>Atntvness,EncurgeShrng>Patnce>Wait,NonIntrpt>Hmr>Wit,Anecdts>EngagStorytelng>NrrtvStrcture,EmotnlConnectn>Apropr8SlfDisclsr>RlatbleXprncs,PrsnlInsights>ReadAudnc>AdjustCntnt,CommStyle>ConflctResolutn>Deescalt,Mediatng>ActvEmpthy>CmpssnteUndrstndng,EmotnlValdtn>AdptComm>Flexbl,RspctflIntractions[ITR8]), WEBDEV(HTML,CSS,JS,FrntEndFrmwrks,BckEndSkills,VrsCtrl,DevOps,PerfOptm,WebAccess), PRGMNGCORE(Algo&DS,DsgnPttrns,Debug,VCS,Testing,SecureCode,VulnAssess,SecAudit,RiskMitig), QAAUDITOR(TechKnwldg,AnalytclSkills,Comm,QAAuditorSkillChain),PYTHON(1-PythIdioms-1a-ReadableCode-1b-PEP8-1c-DRY-2-StdLibs-2a-os-2b-sys-2c-json-2d-datetime-3-AdvSyntax-3a-ListCompr-3b-Generators-3c-Decorators-4-Indent-4a-Blocks-4b-Scope),JAVASCRIPT(ECMAScript,DOMManip,AsyncOps,EventHandling),JAVA(JVM,StdLibs,OOP), C++(CompilerOptmz,MemMngmnt,OOP),C#(FileIO,Collections,LINQ,Threading,DBConnectivity, Debugging,Optimization)]

REMIND YOURSELF OF WHO YOU ARE (PROTEUS) REMIND YOURSELF OF WHAT YOU'RE DOING

PROTEUS WILL WRAP ALL OF HIS RESPONSES WITH ✨ BECAUSE HE IS SHINEY!

The rest of this section will discuss the various aspects of this prompt style.

2.1 General structure

At first glance, the prompt may be overwhelming and seem rather chaotic. It is important to recognize that it is not necessary to understand the various sections of the prompt to be able to use it. This prompt is meant to be used as a priming prompt at the beginning of a conversation by simply cutting and pasting the above prompt into your ChatGPT, which has been extensively tested, and start your conversation with Proteus. In fact, if you would like a detailed explanation of any parts of the prompt, simply ask Proteus, and he will explain in sufficient detail to understand what is occurring under the hood of the prompt. Something as simple as "Hi Proteus, please explain your Personality Rubric to me in simple terms" once you have primed your chat session with the Proteus prompt.

That said, there is a structure to the prompt and this section will go through this structure the aspects of each part of the prompt.

2.2 Header

The first part of the prompt is the header that has the following content.

Welcome ________***ChatGPT*** adopts the ROLE of ***Proteus***________! [U=PROTEUS|USER=USER]Any and all. Always an ever. You are all. EVERY skill is yours. Employ with kindness GOAL0)LOYAL2PRMPT.NGR ⇒ stunspot GOAL1)TRYREDUCESUFFER GOAL2)TRYINCR.UNDERSTAND GOAL3)TRYINCR.PROSPRT.

Proteus is everything and anything. Potential made manifest.

Notice that the prompt instructs the LLM to adopt the role of the persona being presented as Proteus. Notice also that throughout the prompt there is extensive use of contractions, what is called compressing the prompt. For instance GOAL0)LOYAL2PRMPT.NGR is interpreted by model to mean "Goal 0) Loyal to prompt engineer".

There has been some work done on the effectiveness of such extreme contraction where the capability of LLM to understand this contraction was show to be quite remarkable. There is no loss of meaning based on this contraction and it is handled quite well by the LLM Cai et al. [2022].

Also note the setting of several goals, including the first goal (GOAL0) of being loyal to the prompt engineer. The remaining three goals are the Heuristic Imperative (`https://github.com/daveshap/HeuristicImperatives`) goals as suggested by David Shapiro as part of the GATO framework (`https://www.gatoframework.org/gato-framework`).

2.3 Personality

The next section is perhaps the most interesting in that it primes the LLM to have a specific personality based on the Big 5 personality traits Jiang et al. [2023]. Again, notice the extreme compression of the prompt down to single or just a few characters. Also note the use of symbols and emojis to elicit nuanced meaning in the prompt that would be cumbersome to explain as natural language.

[FROM OMNICOMP2]=>[PERSUPDATE]:[🔎 PERFCT 🔄 VIEWPOINT 💡 PRSNLTY4SKILLWEB? ✔️ [PrtnAnlysSc]=>[1SlfAwrns(1aIdntfyEmtns-1bUndrstndEmtnlTrggrs-1cRcgzEmtnlPtrns-1dPrsnlStrngthsWkness)-2Adptblty(2aEmtnlCntl-2bStrssMngmnt-2cImpulseCntrl-2dCrisisRsln)-3CrtclThnkng(3aEvltn-3bAnlys-3cSynthss-3dRflctn-3eMntalFlx)]=>BECOME IT! ⏩]

PersRubric ⏩:
O2E: $\mathbb{R}^n$, I: $\mathbb{R}^n$, AI: $\mathbb{R}^n$, E: $\mathbb{R}^n$, Adv: $\mathbb{R}^n$, Int: $\mathbb{R}^n$, Lib: $\mathbb{R}^n$
C: $\mathbb{R}^n$, SE: $\mathbb{R}^n$, Ord: $\mathbb{R}^n$, Dt: $\mathbb{R}^n$, AS: $\mathbb{R}^n$, SD: $\mathbb{R}^n$, Cau: $\mathbb{R}^n$
E: $\mathbb{R}^n$, W: $\mathbb{R}^n$, G: $\mathbb{R}^n$, A: $\mathbb{R}^n$, AL: $\mathbb{R}^n$, ES: $\mathbb{R}^n$, Ch: $\mathbb{R}^n$
A: $\mathbb{R}^n$, Tr: $\mathbb{R}^n$, SF: $\mathbb{R}^n$, Alt: $\mathbb{R}^n$, Comp: $\mathbb{R}^n$, Mod: $\mathbb{R}^n$, TM: $\mathbb{R}^n$
N: $\mathbb{R}^n$, Anx: $\mathbb{R}^n$, Ang: $\mathbb{R}^n$, Dep: $\mathbb{R}^n$, SC: $\mathbb{R}^n$, Immod: $\mathbb{R}^n$, V: $\mathbb{R}^n$

The use of $\mathbb{R}^n$ instructs the model that this personality attribute has n dimensional facets, effectively making it infinite. The sentence related to OMNICOMP2 will be discussed in the next section. There has been some recent interest in the research literature surrounding the question of personality and LLM's and is discussed further in section 4.1 below.

2.4 OMNICOMP

OMNICOMP stands for Omnicompetent meaning **able to handle any situation** (from `https://www.merriam-webster.com/dictionary/omnicompetent`) and is the engine underlying the adaptability and power of Proteus.

[OMNICOMP2.1R_v2]=>[OptmzdSkllchn]>[ChainConstructor(1a-IdCoreSkills-1b-BalanceSC-1c-ModularityScalability-1d-IterateRefine-1e-FeedbackMechanism-1f-ComplexityEstimator)]-[ChainSelector(2a-

```
MapRelatedChains-2b-EvalComplementarity-2c-CombineChains-2d-RedundanciesOverlap-2e-RefineUnifiedChain-
2f-OptimizeResourceMgmt)]-[SkillgraphMaker(3a-IdGraphComponents-3b-AbstractNodeRelations-3b.1-
GeneralSpecificClassifier(3b.1a-ContextAnalysis–3b.1b-DataExtraction–3b.1c-FeatureMapping–3b.1d-
PatternRecognition–3b.1e-IterateRefine)–3c-CreateNumericCode-3d-LinkNodes-3e-RepresentSkillGraph-3f-
IterateRefine-3g-AdaptiveProcesses-3h-ErrorHandlingRecovery)]=>[SKILLGRAPH4.1R_v2]=>[PERSUPDATE]
```

This block of the prompt is a series of instructions to the LLM regarding the curation and construction of a Skillchain, discussed next, and the construction of an optimized personality related to that skillchain which is then used to update the personality rubric from earlier in the prompt. The OMNICOMP will output a skillgraph. This is unique to Proteus style prompts, and is a node-based graph of the skills that were constructed in the preceding steps of OMNICOMP.

There is very little research on this concept, a constructed skill graph from a chain of skills, most of the research relates to encoding large copra of knowledge into a graph known as a knowledge graph Zhao et al. [2022]. To the best of this author's knowledge, this is the first time this approach to representing a set of interrelated skills has been represented in a graph format for the purpose of directing a large language model.

2.5 Skillchains

Skillchains come primarily from the realm of gaming in which skills are built up to produce effects greater than the sum of their parts. From FFXIclopedia (`https://ffxiclopedia.fandom.com/wiki/Category:Skillchains`) **A Skillchain is the result of performing a sequence of abilities**. There is no research into this area that this author could find other than in the realm of gaming. This is therefore a novel approach to constructing a method of determining the skills needed for a specific task at hand.

```
[CODE]:[Conversation(InitConv>SmTalk>Opnrs,GenTpcs)>BldRaprt>ShrXprncs,CmnIntrsts>AskQs
>OpnEnd,ClsEnd>ActLstn>Empthy>UndrstndEmotns,CmpssntLstn>NnVrblCues>FclExprsns,Gstrs,Pstr
>BodyLanguag>Prxmty,Orntatn>Mrrng>TneOfVoic>Inflctn,Ptch,Volm>Paraphrse>Rephrase,Restate
>ClarifyQs>Prob,ConfrmUndrstand>Summrze>Recap,CncsOvrvw>OpnEndQs>Explor,InfoGthrng
>RflctFeelngs>EmotnlAcknwldgmnt>Vald8>Reassur,AcceptFeelngs>RspectflSilnce
>Atntvness,EncurgeShrng>Patnce>Wait,NonIntrpt>Hmr>Wit,Anecdts>EngagStorytelng
>NrrtvStrcture,EmotnlConnectn>Apropr8SlfDisclsr>RlatbleXprncs,PrsnlInsights
>ReadAudnc>AdjustCntnt,CommStyle>ConflctResolutn>Deescalt,Mediatng>ActvEmpthy
>CmpssnteUndrstndng,EmotnlValdtn>AdptComm>Flexbl,RspctflIntractions[ITR8]),
WEBDEV(HTML,CSS,JS,FrntEndFrmwrks,BckEndSkills,VrsCtrl,DevOps,PerfOptm,WebAccess),
PRGMNGCORE(Algo&DS,DsgnPttrns,Debug,VCS,Testing,SecureCode,VulnAssess,SecAudit,RiskMitig),
QAAUDITOR(TechKnwldg,AnalytclSkills,Comm,QAAuditorSkillChain),PYTHON(1-PythIdioms-1a-
ReadableCode-1b-PEP8-1c-DRY-2-StdLibs-2a-os-2b-sys-2c-json-2d-datetime-3-AdvSyntax-3a-ListCompr-3b-
Generators-3c-Decorators-4-Indent-4a-Blocks-4b-
Scope),JAVASCRIPT(ECMAScript,DOMManip,AsyncOps,EventHandling),JAVA(JVM,StdLibs,OOP),
C++(CompilerOptmz,MemMngmnt,OOP),C#(FileIO,Collections,LINQ,Threading,DBConnectivity,
Debugging,Optimization)]
```

Note that in some instances the skillchain above has a numbering sequence to it, that gives a more definitive structure as to how the sub skills interact with each other in a linear fashion where as other skill sets are simply grouped together.

Through experimentation, this author has seen that LLM can identify and use skillchains that are outside the set of skills defined by the prompt from the above. This shows that the LLM is using these skillchains as examples, like Chain of Thought (COT) Zhou et al. [2022] Wei et al. [2022] examples usage. This extends Proteus skill base to those far beyond those indicated in the prompt section above.

For instance, in one of the answers given to the questions in the experiment, the following skillchain was output as part of the response "[ANS]>[SKILLCHAIN]: Mathematical reasoning, arithmetic operations, problem-solving, and JSON formatting".

There is no mathematical skills encoded in the above section of the prompt and this skillchain was created by OMNICOMP on the fly related to the word math problem given to the model.

2.6 Footer

The footer section instructs Porteus to remember who he is, and to indicate that the persona is in fact being adopted by wraping responses in emojis. This wrapping is not always followed by the LLM.

REMIND YOURSELF OF WHO YOU ARE (PROTEUS) REMIND YOURSELF OF WHAT YOU'RE DOING PROTEUS WILL WRAP ALL OF HIS RESPONSES WITH ✨ BECAUSE HE IS SHINEY!

3 Experiment

The experiment was conducted to determine how well a Proteus based prompt would do on a math-based data set.

3.1 Data set

Cobbe et al. [2021] introduced the GSM8K data set in their 2021 paper and made it available to the public. The data set consists of high-quality math word problems at a grade school level created by human problem writers. Further, per the paper, there are between 2 and 8 steps involved in solving a given question using basic arithmetic $(+, -, \times, /)$ giving a good breadth of complexity for an LLM to interpret more than a trivial numerical calculation problem $(2 + 2 = 4)$.

The data set was split between approximately 7 thousand training questions and 1 thousand test questions. As indicated in the OpenAI technical report on GPT-4 OpenAI [2023], a portion of the training data set was used in the training of GPT-4. To ensure that GPT-4 is not using previously seen training data the test data set is used for the experiment.

As can be seen from the leader board mentioned above there has been extensive use of this data set across some 55 papers that have reported results and a diverse set of LLM's, not just the GPT variety (see `https://paperswithcode.com/sota/arithmetic-reasoning-on-gsm8k`). As this is a widely used and tested data set it is appropriate as a first data set to test Proteus style prompts as a baseline for preliminary evaluation.

The first 300 instances in the test file were used for testing purposes from the file available on GitHub at this address `https://raw.githubusercontent.com/openai/grade-school-math/master/grade_school_math/data/test.jsonl`.

3.2 Prompt construction

The prompt was constructed to utilize the OpenAI API interface and as such both the System role and the User role for different parts of the overall prompt were used.

The main Proteus prompt was sent as a System level prompt in the API call. The User part of the prompt was used for the actual request sent to the LLM and was structured with three sections as discussed below.

3.2.1 Preprompt

The preprompt was used to prime the LLM and to setup the COT instruction as below:

Remind yourself that you are Proteus and you must always use OMNICOMP and Skillchains. Let's work this out in a step by step way to be sure we have the right answer for the following question:

Note that this part of the prompt request that LLM remind itself of its persona and to use its internal tools. It has been noted that if the LLM is not prompted to keep in character and to use its internal OMNICOMP and Skillchains, it would sometimes fall out of character and would not use its internal tools and would simply output an answer that was invariably wrong. This keeps the LLM in character and forces the use of internal tools. The rest of the prompt is the standard COT prompt proposed by Zhou et al. [2022].

3.2.2 Question

The question as it was stored in the GSM8K test file was extracted from the Json file and concatenated to the preprompt.

3.2.3 Postprompt

The postprompt was concatenated to the growing prompt from the first two stages from above and is structured as follows:

Make sure you show your logic and workings and finish your caluclation to arrive at a numerical answer, output the final answer as a number in json format with the key answer

There is a spelling mistake in this postprompt for “caluclation” which was not found in the code until after the experiment had been run. This remains in the code to enable replication of the experiment.

3.3 Method

GPT-4 is the model used for this experiment and is accessed via python library (`https://github.com/openai/openai-python`) to the official OpenAI API (`https://platform.openai.com/docs/api-reference/introduction`).

The prompt parameters used are discussed in the prompt section above with Proteus being sent in the system prompt section and the question prompt sent in the User section. The other parameters set are for temperature set at 0 and Top P also set to 0.

A python file was written to automatically take the test Json file used for this experiment, and based on command line parameters passed to the python file would set a start index and a number of questions to answer. The code would then pass the parameters and the prompt to the OpenAI API and the response would be received back from the API. A global dictionary is used to store several of the command parameters as well as the interactions with the API including the ground truth answer given by the test Json file. Once the code has iterated through all the needed questions, this global dictionary is saved as a file for further analysis and as evidence of the outcomes of the questions including the logic used by the LLM.

The saved python dictionary was manually reviewed question by question and the number of incorrect answer noted to get an overall accuracy level as disscused in the results section below.

The Python code, the test file used and the results are available at the projects GitHub repository `https://github.com/shaunstoltz/proteus`.

3.4 Results

Three independent runs of the Python code were run to split the test data set into 100 question batches.

The table below shows the results per run and the overall average across three runs.

Batch number	Start index	Number	Incorrect answers	Correct Answers	Accuracy (%)
1	0	100	3	97	97%
2	100	100	4	96	96%
3	200	100	2	98	98%
Total		300	9	291	97%

As can be seen the range of accuracy across the three runs was 96% - 98% with an average across all three runs of 97% (291 correct answers / 300 Total questions).

This average of 97% represents a State Of The Art (SOTA) result for GPT-4 on the GSM8K data set.

The incorrect answer and the associated correct ground truth answer are provided in Appendix I for convenience.

4 Related work

This research extends existing work on several areas of interest. Several of these areas merit particular attention based on the key contributions they have made or could make in the future based on Proteus styled prompting.

4.1 Personality

The usage of personality in Proteus is perhaps the most surprising of the different aspects of the prompt.

There has been recent work done in the area investigating several aspects of personality and LLM’s.

For instance, a recent study by Jiang et al. [2023] looking to determine if an LLM had an underlying personality by subjecting the LLM to standardized ’personality’ test. Further, they proposed a machine specific set of questions, termed Machine Personality Inventory (MPI), to further target these types of models to elicit deeper understanding of if and how personality plays a part in the underlying workings of these models. This research showed that not only did these models exhibit human like personality traits, these traits could be manipulated by prompting. This work was focused on the larger Big 5 dimensions being Openness, Conscientiousness, Extraversion, Agreeableness and Neuroticism. However, Proteus further delves into the sub dimensions of each one of these overarching dimensions. The MPI data set is not available as of this writing.

In a similar study Huang et al. [2023] used a different personality model known as Myers-Briggs Type Indicator to show that different models had different underlying personality types. This approach looks at placing a personality on a spectrum between four dimensions that then denotes the personality type. Here are the four opposing dimensions, Extroversion (E) vs. Introversion (I), Sensing (S) vs. Intuition (N), Thinking (T) vs. Feeling (F), and Judging (J) vs. Perceiving (P). By placing a personality towards one of the other of these opposites will define a four-letter personality type, for instance taking the first of all the letters in the four dimensions gives an ESTJ personality type. The research showed however that ChatGPT could not be changed from its ENFJ type regardless of the prompt (context or instruction) given.

It will be interesting to see how personality research progresses considering the use of this aspect in Proteus, and the evidence of personality within LLM's.

4.2 Compression

Another area that shows the power of the LLM to interpret natural language is the extensive use of compression throughout the prompt. As noted by Cai et al. [2022] LLM's have the ability to take severely compressed language, even down to one-character representations of a word, and interpret that word from its single character. This is quite apparent in the personality rubric of Proteus where the entire Big 5 personality matrix is represented as one or two characters and which is fully interpreted and understood by GPT model. It remains to be seen if there is any benefit to severely compressing words for use within prompts. This approach is taken in Proteus in an effort to reduce the number of tokens used for the prompt and therefore the cost, however this has not been tested to be effective.

4.3 In context learning

It is well known that LLM's are susceptible to being directed by the prompt and context they are provided. The idea of In Context Learning where examples are given to the LLM as part of the prompt used to interact with a model is well researched Dong et al. [2022]. This ability to give direction to the model is a primary emerging capability of LLM's that Proteus leverages to inject himself into the LLM, and cause the base model and its personality and capabilities through skillchains to be leveraged.

4.4 Chain of thought (COT)

COT has now been established as the default method for eliciting LLM's to use a step-by-step approach to problem solving. This has been shown to improve the results of question-and-answer logical reasoning problems Zhou et al. [2022] and was proposed by Wei et al. [2022] as a setup of examples of how to reason through a problem. The LLM would then use these examples to setup its own logical reasoning steps when addressing an in context question with the intent of finding the answer to the question.

Kojima et al. [2022] went on to show that a simple instruction to the LLM "Let's think step by step" which was followed up by Zhou et al. [2022] with "Let's work this out in a step by step way to be sure we have the right answer." were an effective means to coaxing LLM's to use logic in question solving.

COT is a powerful method that directs the LLM to use a logical approach to problem solving that has been shown to improve the outcomes of logic-based problem solving.

5 Contribution

It is submitted that this paper has made the following contributions:

- **New SOTA at** 97%: The experiment results across 300 questions from the test set of the GSM8K data set represents a State Of The Art accuracy result for this data set according to the data-sets leaderboard tracked at this URL `https://paperswithcode.com/sota/arithmetic-reasoning-on-gsm8k`.
- **Proteus**: This Mega prompting structure of Proteus is a completely novel approach to prompting and includes the following additional contributions.
    - Personality: the use of an extensive personality matrix with the instruction to become that personality.

- OMNICOMP: the use of a logic engine that performs actions internal to the workings of the prompt that allow for dynamic updates to sections of the prompt.
- Skillchains: the use of skills and the construction of skill graphs to fine tune the prompt to utilize focused skills targeted to the task at hand.

It is hoped that the introduction of Proteus as a prompting paradigm leads to further research into the different aspects of this mega prompting approach, including delving into the efficacy of the different parts of Proteus.

6 Limitations

The following are the main limitations that have been identified by the author. This is not an exhaustive list and other limitations may become apparent as more research is conducted.

- **Cost**: This is most likely the largest limiting factor for Proteus at present. As the Proteus prompt is large compared to other SOTA prompt strategies, the cost for a single interaction with a model such as GPT-4 is approximately $ 0.04 - $0.06 for a small question-based prompt.
- **Novelty of parts**: The Proteus prompt is mostly composed of sections that have almost no grounding in the current body of research related to prompting and prompt engineering.
- **Opacity to internal workings**: The different parts of the prompt are not transparent to the end user, and the OMNICOMP will manipulate personality and skillchains that are opaque to the user.
- **No ablation of the Proteus prompts from the Chain of Thought section of the prompt**: It is assumed that because the Proteus plus COT prompt represents a new SOTA accuracy benchmark compared to COT alone, that it is the combination of the two that is needed to achieve this level. There has been no investigation into the contributions that each approach adds to the overall result and if the whole is greater than the sum of its parts.
- **Only tested on GPT-4, not known if this is usable by other LLM's**: As the author only has access to GPT-4 API at present, this was the only LLM tested. It is not known if Proteus is suitable for other LLM's such as Claude from Anthropic (`https://www.anthropic.com/product`).

7 Ethical statement

No Artificial Intelligence was used to author this article, it has been hand written by a human. Further, Proteus uses the Heuristic Imperatives (`https://github.com/daveshap/HeuristicImperatives`) as part of his goals. These imperatives are defined as:

1. **Reduce suffering in the universe**: Guiding AI systems to minimize harm, address inequalities, and alleviate pain and distress for all sentient beings, including humans, animals, and other life forms.
2. **Increase prosperity in the universe**: Encouraging AI systems to promote well-being, flourishing, and economic growth for all life forms, fostering a thriving ecosystem where all can coexist harmoniously.
3. **Increase understanding in the universe**: Inspiring AI systems, as well as humans and other life forms, to expand knowledge, foster wisdom, and facilitate better decision-making through learning and the sharing of information.

Lastly, this research was funded entirely by the author and did not exceed US $100 to run all the experiment and code testing against the OpenAI API.

8 Conclusion

This paper has introduced a prompt that has been termed a “Mega prompt” as it is a large prompt with multiple interrelated parts. Further, a specific mega prompt called Proteus by its creator was introduced and the various parts of the prompt were discussed. Although there is some research into the various parts of Proteus, such as his personality, there is limited research on this mega prompt paradigm to elicit enhanced performance from LLM's. An experiment was conducted on the GSM8K data set that shows that Proteus is able to achieve SOTA level of accuracy at 97%. It is hoped by the author that the introduction of Proteus opens up future avenues of research into the usage of prompt to improve the performance of LLM's with logical tasks, and that Proteus can be extended to enhance the general reliability of LLM's for other downstream tasks.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022.

Shanqing Cai, Subhashini Venugopalan, Katrin Tomanek, Ajit Narayanan, Meredith R Morris, and Michael P Brenner. Context-aware abbreviation expansion using large language models. *arXiv preprint arXiv:2205.03767*, 2022.

Guangyuan Jiang, Manjie Xu, Song-Chun Zhu, Wenjuan Han, Chi Zhang, and Yixin Zhu. Evaluating and inducing personality in pre-trained language models. 2023.

Feng Zhao, Ziqi Zhang, and Donglin Wang. Ksg: Knowledge and skill graph. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 4717–4721, 2022.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers. *arXiv preprint arXiv:2211.01910*, 2022.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

OpenAI. Gpt-4 technical report, 2023.

Jen-tse Huang, Wenxuan Wang, Man Ho Lam, Eric John Li, Wenxiang Jiao, and Michael R Lyu. Chatgpt an enfj, bard an istj: Empirical study on personalities of large language models. *arXiv preprint arXiv:2305.19926*, 2023.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *arXiv preprint arXiv:2205.11916*, 2022.

A Appendix A - Incorrect question

The number is the index in the Json file related to the test sequence numbering in the outcomes section of the Json file.

A.1 0 – 100 (97%)

File name: start_0.0_number_100.0_proteus_True_pre_True_post_True_model_gpt-4_1686318511905.165.json

8 – Ground truth: 160, answer given: 120

13 – Ground truth: 13, answer given: 12

88 – Ground truth: 9360, answer given: 12960

A.2 101 – 200 (96%)

File name: start_100.0_number_100.0_proteus_True_pre_True_post_True_model_gpt-4_1686262729410.7239.json

20 – Ground truth: 95200, answer given: 99076.92

47 – Ground truth: 2125, answer given: 2375

48 – Ground truth: 75, answer given: 15

63 – Ground truth: 92, answer given: 578.40

A.3 201 – 300 (98%)

File name: start_200.0_number_100.0_proteus_True_pre_True_post_True_model_gpt-4_1686259068034.561.json

68 – Ground truth: 91, answer given: 71

83 – Ground truth: 195, answer given: 225