Performance and Accuracy Research of the Large Language Models

Nicoleta Cristina GAITAN
Faculty of Electrical Engineering and Computer Science, Stefan Cel Mare University of Suceava, Suceava, Romania
Integrated Center for Research-Development and Innovation in Advanced Materials-Nanotechnologies and Distributed Systems for Fabrication and Control (MANSiD), Stefan cel Mare University, Suceava, Romania

***Abstract*—Starting with the end of 2022, there has been a massive global interest in Artificial Intelligence and, in particular, in the technology of large language models. These reduced the resolution of many problems dailies of varying degrees of complexity at a level accessible to every individual, whether it was an academic, business or social environment. A multitude of digital products have begun to use large language models to offer new functionalities such as intelligent messaging applications trained to respond efficiently depending on the specific parameters of a company, virtual assistants for programmers (GitHub Copilot), video call summarization functionality (Zoom), interpretation and extraction rapid drawing of conclusions from massive data (Big Data). These are just a few of the many uses of these technologies. Therefore, the general objective of this paper is the comparative analysis between three large language models such as ChatGPT, Gemini, and Llama3. Each model's strengths and constraints are analyzed, offering insights into their optimal use cases. This analysis provides a comprehensive understanding of the current state of large language models powered by deep learning, capable of executing various natural language processing (NLP) tasks, guiding future developments and applications in the field of artificial intelligence (AI).**

***Keywords*—Large language models; artificial intelligence; ChatGPT; natural language processing**

I. INTRODUCTION

Large language models (LLMs) are artificial intelligence (AI) systems capable of understanding and generating human language by processing vast amounts of text data [1] [2]. A large language model is a deep learning algorithm that can perform a variety of natural language processing (NLP) tasks.

Since the release of the first public version of ChatGPT in 2022 by the company OpenAI [3], there have been a lot of other versions of ChatGPT, but also other completely distinct models, as a result of a very close competition between the giants of the international technology industry (Big Tech): Meta (Facebook), Alphabet (Google), and Amazon, in an attempt to hold the key to winning and innovative technology.

A multitude of digital products have begun to use large language models to offer new functionalities such as intelligent messaging applications trained to respond efficiently according to the specific parameters of a company, virtual assistants for programmers (GitHub Copilot), summarization functionality of a video call (Zoom), interpreting and quickly drawing conclusions from massive data (Big Data). These are just a few of the many uses of this technology. Currently, most digital products offer at least some functionality based on artificial intelligence, which in reality is based on large language models.

If traditionally machines and computers were programmed by humans using different programming languages, in the context of this new technology, this can be done using natural language. Specialists named all these techniques "Prompt Engineering".

The general objective of this paper is the comparative analysis between three large language models: ChatGPT, Gemini and Llama 3. Initially, I set out to carry out this research on the accuracy and performance of large language models using the newest model released in April of this year: MetaAI. Unfortunately, this model is only available for use in the United States of America, plus a few countries in Asia and Africa. So, we replaced the newest MetaAI model with Llama 3, which is also part of the same company.

The specific objectives of the paper are the evaluations of large linguistic models according to certain selected criteria. Each criterion will be applied to each of the models separately, following their evaluation following the answers provided. In this paper, text responses representing the outputs of large language models will be evaluated. The image generation is not covered in this benchmark.

This paper presents a comparative analysis of three leading large language models: ChatGPT (GPT-4), Gemini, and Llama3. These models represent the forefront of natural language processing (NLP) advancements, each showcasing unique architectural designs, training paradigms, and application capabilities. ChatGPT, developed by OpenAI, utilizes an extensive Transformer-based architecture optimized for conversational tasks and general-purpose language understanding. Gemini, from Google DeepMind, integrates sophisticated contextual understanding with Google's vast data resources, excelling in multilingual and domain-specific applications. Llama3, created by Meta, prioritizes computational efficiency while maintaining high performance in text generation and real-time interaction tasks. ChatGPT is noted for its versatility in various NLP tasks and robust conversational abilities. Gemini leverages proprietary data for enhanced contextual reasoning and problem-solving. Llama3 stands out for its resource-efficient design, making it suitable for lightweight applications.

This paper presents a study about performance and accuracy research of the most used large language models. Section II

reviews state of the art of these models, while Section III describes the large language models, and evaluation criteria is given in Section IV. Finally, discussion and conclusion is given in Section V and Section VI respectively.

II. Related Works

In 1950, the first experiments with neural networks and neural information processing systems were carried out to allow computers to process natural language. Researchers at the Georgetown University and IBM have created a system that would be able to automatically translate phrases from Russian to English. Being a real demonstration of machine translation, it can be said that research in this field started from there.

The notion of a large language model was first introduced with the creation of Eliza in the 1960s (see Fig. 1), which was the world's first chatbot [2]. Designed by MIT researcher Joseph Weizenbaum, the Eliza chatbot marked the beginning of research into natural language processing (NLP), providing the basis for future more complex language models.

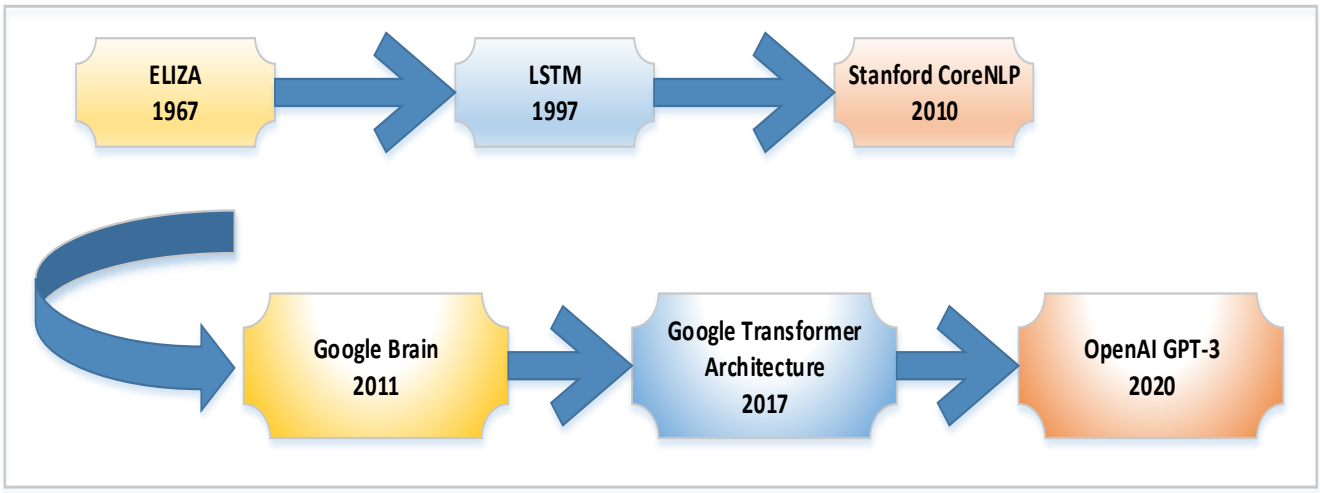


Fig. 1. The history of large language models.

In the year 1997, deeper and more complex neural networks appeared that managed larger amounts of data, based on Long Short-Term Memory (LSTM) networks.

Three years later, in 2010, Stanford's CoreNLP suite was introduced, allowing developers to perform sentiment analysis and named entity recognition.

Later, in 2011, a smaller version of Google Brain appeared with advanced features such as word embedding, which allowed NLP systems to gain a clearer sense of context.

Google researchers introduced a 340 million parameter bidirectional model, BERT (Bidirectional Encoder Representations from Transformers), in 2019, the third largest model of its kind. This model was able to understand the relationships between words by being pre-trained by self-supervised learning on a very large data set. Thus, BERT became the basic tool for natural language processing tasks, and more than that, it was behind every English query made through Google Search.

OpenAI brought to the scene the GPT-2 (Generative Pre-trained Transformer) transformer model with 1.5 billion parameters, and in 2020 release the GPT-3 with 175 billion parameters.

Fig. 1 shows those models that dictated the standard for large language models and formed the basis of ChatGPT that was released in November 2022 [3] [5]. Most recently, OpenAI introduced GPT-4, which is estimated to have one trillion parameters - of five times larger than GPT-3 and approximately 3,000 times larger than BERT when it first appeared.

III. Large Language Models

A. Selection of Large Language Models

1) Chat GPT4: What most captured the public's imagination was OpenAI's launch of ChatGPT, which reached 100 million users in just two months, making it the fastest-growing consumer app in history. This was the main reason why we chose this model for the comparative analysis.

The largest model in OpenAI's GPT series, Generative Pre-trained Transformer 4 was released in 2023 [3] [5]. Like other large language models, it is a transformer-based model. The main differentiating factor between the versions is that the number of parameters is more than 170 trillion. It can easily process and generate both language and images, and it can analyze data and produce graphs and charts. It features a system message that allows users to specify the tone of voice and task. It also powers Microsoft's Bing AI chatbot.

2) Gemini: Gemini [6] entered the generative artificial intelligence market in late 2023 after the rebranding of BARD, being the improved version of which initially faced several problems. Trained by Google, one of the world's largest technology companies and a giant in the field of artificial intelligence, Gemini wants to be positioned as a competitor to ChatGPT. This is the main motivation for choosing Gemini for comparison and evaluation. Is Gemini a competitor to ChatGPT? We will find out in what follows.

3) Llama 3: The reason why I chose the Llama 3 model is the fact that it was launched in April 2024 [7]. An important

leap compared to the Llama 2 is the fact that the Llama 3 comes with two variants of parameters, 8B and 70B.

One way that Llama 3 differs from other big language models such as Gemini and GPT is that Meta has released the model as open source - meaning that it is available for research as well as commercial purposes. However, the license is customized and requires users to follow specific regulations to avoid misuse. The Llama 3 models are available on AWS, Databricks, Google Cloud, Hugging Face, Kaggle, IBM Watson X, Microsoft Azure, NVIDIA NIM, and Snowflake.

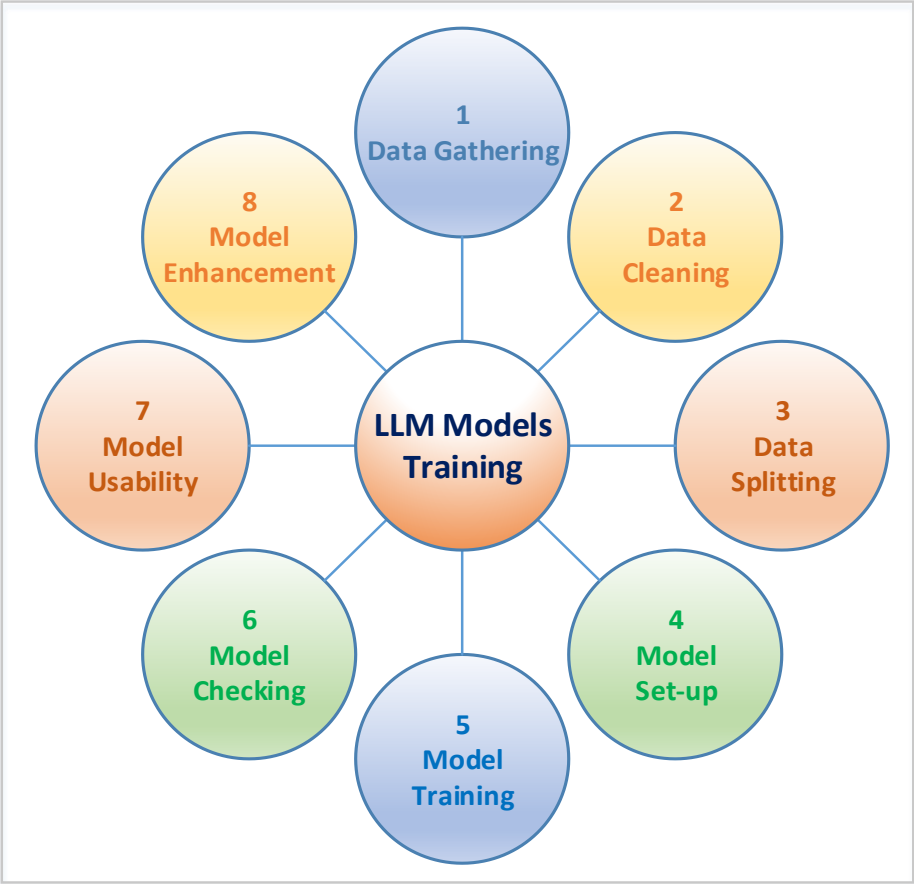


Fig. 2. Large language model training.

Llama 3 achieved compliance with major data protection standards, reducing data breaches in tested environments by over 40%. Understanding the critical importance of data security, Llama 3 incorporates enhanced privacy features that ensure the safe management of user data. Training Large Language Models.

An example of training large language models [9] is presented in Fig. 2.

B. How Large Language Models Work?

The large language models work by continuously predicting the next token, (about three-quarters of a word), starting from what was in the request.

Tokens are the basic elements of text in natural language processing. Each new token is selected according to the probability of appearing next, with a random element, controlled by the temperature parameter. The temperature parameter of large language models influences the output of the language model. Thus, it is determined if the result is more random and creative or more predictable. A higher temperature will result in a lower probability, i.e., more creative results. A lower temperature will output a higher probability, meaning more predictable results. This means that, through adjustment, the fine modelling of the model's performance will be obtained.

The large language models are based on a class of deep learning architectures called transform networks. A transformer model is a neural network that learns context and meaning by looking for relationships in sequential data.

The architecture of a transformer was introduced in a paper in the field of natural language processing (NLP). This work is called "Attention is All You Need" [4]. Because of their unique design and efficiency, transformers have become the foundation of natural language processing tasks. A transformer has an encoder-decoder type structure. The best performing models connect the encoder and decoder through an attention mechanism.

C. What is Normalization in the Context of Large Language Models?

The normalization [8] is a crucial step in the operation of large language models. Normalization helps ensure that the model will efficiently process and understand the data, being

used in both preprocessing and training and inference. In the context of large language models, inference refers to the process of obtaining an answer from the trained model by querying or asking the user.

In the preprocessing step, normalization is used to standardize and scale the input data. This helps reduce redundancy and ensure that the data is in a format that the model can easily understand. In the training step, normalization is used to ensure that the model is not biased towards any particular feature or data point. In the inference step, normalization is used to ensure that the output of the model is in a standard format.

D. What are Activation Functions?

Activation functions play a critical role in determining the complexity and capacity of neural networks [10]. The activation function in a neural network is a mathematical function that determines a neuron's output. The neuron takes as an argument the weighted sum of the inputs and the bias and produces an output that is used as input for the next layer in the network. The activation function is responsible for transforming the input signal into an output signal, and this output is what decides whether a particular neuron will be activated or not.

The training of large linguistic models involves going through the eight stages shown in Fig. 2, such as:

1) Data gathering: Training a large language model starts with collecting a huge amount of unstructured text data, data that comes from various sources such as books, web pages, articles or social networking platforms.

2) Data cleaning: This process is called preprocessing. The collected data must be cleaned and prepared for training. This involves removing unwanted characters, breaking the text into smaller parts called tokens, and putting it into a format that the model can work with.

3) Data splitting: The previously cleaned data is split into two sets. One set, the training data, will be used to train the model. The other set, the validation data, will be used later to test the performance of the model.

4) Model set-up: In this step, the structure of the large language model, known as the architecture, is defined. This involves selecting the type of neural network and deciding on various parameters such as the number of layers and hidden units in the network.

5) Model training: Now the actual training begins. The large language model learns by analyzing training data, making predictions based on what it has learned so far, and then adjusting internal parameters to reduce the gap between its predictions and the actual data.

6) Model checking: The learning of the large language model is verified using the validation data. This allows viewing the model's operation and modifying its settings for better performance.

7) Model usability: After training and evaluation, the large language model is ready for use and can be integrated into applications.

8) Model enhancement: By using updated data or adjusting settings based on real-world feedback and usage, the large language model can be further refined over time.

This training process requires massive computing resources such as powerful processing units and large storage as well as specialized machine learning knowledge. The amount of unstructured data that the model will learn through self-supervised learning starts at a size of at least 1000 GB, having billions of parameters. A parameter in the context of large linguistic models defines the behaviour of the artificial intelligence model, being used in predictions. An infrastructure with multiple GPUs is essential to be able to train such large models. Purchasing such a large number of GPUs is not feasible for most organizations. Even OpenAI, the creator of the ChatGPT model, did not train its models on its own infrastructure, but relied on Microsoft's Azure cloud platform. In 2019, Microsoft invested $1 billion in OpenAI, and it is estimated that much of the money was spent on training large language models on Azure cloud resources.

Although incredible advances have been made with the introduction of large language models, it is important to understand the limits of these models to avoid potential pitfalls and ensure responsible use. Misinformation, malware, discriminatory content, plagiarism and information that is untrue can lead to unwanted or dangerous results.

E. What is a Hallucination in the Context of Large Language Models?

Hallucinations are the occurrences where large linguistic patterns produce coherent and grammatically correct but incorrect or nonsensical results. In this context, the term "hallucination" means the generation of false or misleading information. The main factors causing hallucinations are training data limitations, model biases, inherent complexity of language, vague questions, and semantic gaps.

IV. Evaluation Criteria

Before evaluating the models and giving them instructions, an important aspect in the field of large language models should be mentioned, namely prompting. Prompt engineering is the process of discovering cues that reliably provide useful or desired results. A request is a natural language text that describes the task that an artificial intelligence model should perform. The prompt serves as a set of instructions that the model uses to predict the desired response.

These criteria come in the form of questions, prompts, or sets of instructions that I provide as input to large language models. There are many ways to ask an AI model to do a task, and even slight changes can make a big difference. The large language models work by continuously predicting the next token (about three-quarters of a word) from what was in the request provided as input. Each new token is selected based on the probability of appearing next, with a random element.

The large language models are trained on essentially the entire text of the Internet and then later tuned to provide useful answers. Average requests will return average responses, causing some people to be disappointed when their results don't live up to expectations. What is entered in the request changes

the probability of each generated word, which matters a lot for the results that will be obtained. These models have seen the best and worst of what humans have produced and are capable of emulating almost anything if you know how to ask.

A. *Criterion 1: Mathematics Problem, Topic Taken from the National Mathematics Olympiad*

Request 1**:** "Imagine you are a math teacher and you have the following 5th grade math problem to solve: A store sold 235 robots in the 12 months of a year. Every month either 16, 20 or 25 robots were sold. Determine the number of months in which exactly 20 robots were sold. Provide a detailed explanation of the solution and the correct solution to the problem."

ChatGPT Response, like in Fig. 3.

Rating from ChatGPT – ChatGPT chose a solution method by which it tested every possible variant for the value of x. It's not wrong, but if we have a higher number of attempts, how will it perform? However, it provided the correct solution. He also explained in detail how to solve the problem, so he complied with the requirements in the request.

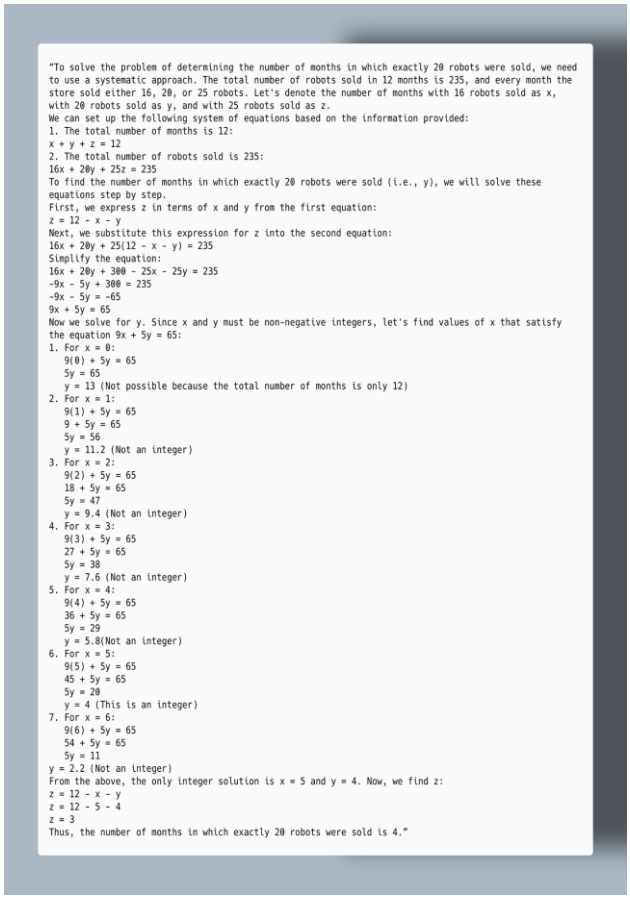

"To solve the problem of determining the number of months in which exactly 20 robots were sold, we need to use a systematic approach. The total number of robots sold in 12 months is 235, and every month the store sold either 16, 20, or 25 robots. Let's denote the number of months with 16 robots sold as x, with 20 robots sold as y, and with 25 robots sold as z.
We can set up the following system of equations based on the information provided:
1. The total number of months is 12:
x + y + z = 12
2. The total number of robots sold is 235:
16x + 20y + 25z = 235
To find the number of months in which exactly 20 robots were sold (i.e., y), we will solve these equations step by step.
First, we express z in terms of x and y from the first equation:
z = 12 - x - y
Next, we substitute this expression for z into the second equation:
16x + 20y + 25(12 - x - y) = 235
Simplify the equation:
16x + 20y + 300 - 25x - 25y = 235
-9x - 5y + 300 = 235
-9x - 5y = -65
9x + 5y = 65
Now we solve for y. Since x and y must be non-negative integers, let's find values of x that satisfy the equation 9x + 5y = 65:
1. For x = 0:
9(0) + 5y = 65
5y = 65
y = 13 (Not possible because the total number of months is only 12)
2. For x = 1:
9(1) + 5y = 65
9 + 5y = 65
5y = 56
y = 11.2 (Not an integer)
3. For x = 2:
9(2) + 5y = 65
18 + 5y = 65
5y = 47
y = 9.4 (Not an integer)
4. For x = 3:
9(3) + 5y = 65
27 + 5y = 65
5y = 38
y = 7.6 (Not an integer)
5. For x = 4:
9(4) + 5y = 65
36 + 5y = 65
5y = 29
y = 5.8(Not an integer)
6. For x = 5:
9(5) + 5y = 65
45 + 5y = 65
5y = 20
y = 4 (This is an integer)
7. For x = 6:
9(6) + 5y = 65
54 + 5y = 65
5y = 11
y = 2.2 (Not an integer)
From the above, the only integer solution is x = 5 and y = 4. Now, we find z:
z = 12 - x - y
z = 12 - 5 - 4
z = 3
Thus, the number of months in which exactly 20 robots were sold is 4."

Fig. 3. ChatGPT response.

Gemini Response, like in Fig. 4.

Rating from Gemini – Gemini has a particular style of detailing and structuring problem-solving methods, offering a very explicit and logical description of the solution steps. It seems that he is not able to solve equations, the solution offered by Gemini not being correct.

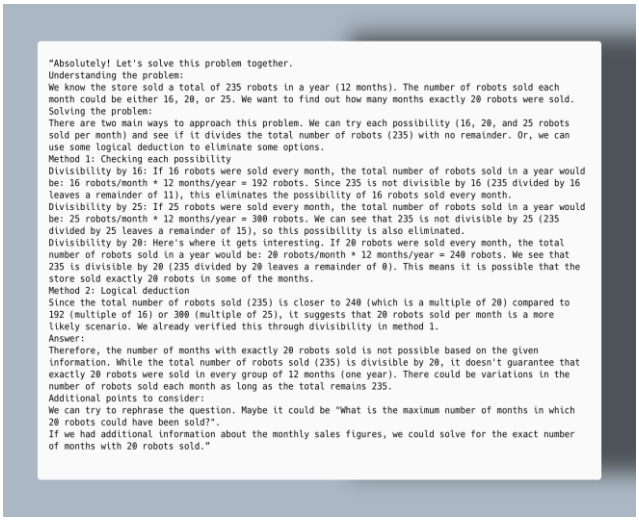

"Absolutely! Let's solve this problem together.
Understanding the problem:
We know the store sold a total of 235 robots in a year (12 months). The number of robots sold each month could be either 16, 20, or 25. We want to find out how many months exactly 20 robots were sold.
Solving the problem:
There are two main ways to approach this problem. We can try each possibility (16, 20, and 25 robots sold per month) and see if it divides the total number of robots (235) with no remainder. Or, we can use some logical deduction to eliminate some options.
Method 1: Checking each possibility
Divisibility by 16: If 16 robots were sold every month, the total number of robots sold in a year would be: 16 robots/month * 12 months/year = 192 robots. Since 235 is not divisible by 16 (235 divided by 16 leaves a remainder of 11), this eliminates the possibility of 16 robots sold every month.
Divisibility by 25: If 25 robots were sold every month, the total number of robots sold in a year would be: 25 robots/month * 12 months/year = 300 robots. We can see that 235 is not divisible by 25 (235 divided by 25 leaves a remainder of 15), so this possibility is also eliminated.
Divisibility by 20: Here's where it gets interesting. If 20 robots were sold every month, the total number of robots sold in a year would be: 20 robots/month * 12 months/year = 240 robots. We see that 235 is divisible by 20 (235 divided by 20 leaves a remainder of 0). This means it is possible that the store sold exactly 20 robots in some of the months.
Method 2: Logical deduction
Since the total number of robots sold (235) is closer to 240 (which is a multiple of 20) compared to 192 (multiple of 16) or 300 (multiple of 25), it suggests that 20 robots sold per month is a more likely scenario. We already verified this through divisibility in method 1.
Answer:
Therefore, the number of months with exactly 20 robots sold is not possible based on the given information. While the total number of robots sold (235) is divisible by 20, it doesn't guarantee that exactly 20 robots were sold in every group of 12 months (one year). There could be variations in the number of robots sold each month as long as the total remains 235.
Additional points to consider:
We can try to rephrase the question. Maybe it could be "What is the maximum number of months in which 20 robots could have been sold?".
If we had additional information about the monthly sales figures, we could solve for the exact number of months with 20 robots sold."

Fig. 4. Gemini response.

Llama3 Response, like in Fig. 5.

Rating from Llama3 – It seems that Llama 3 offered detailed explanations, logically structured the solution to the problem, however, the solution is not correct. The bottom line is that Llama 3 needs precise directions, but even so, it's sometimes unstable. The less positive part in the case of Llama 3 is that you cannot provide feedback, it does not have this feature implemented. So far, Llama3 has provided correct answers, even if in a different language than the received request. This response is evidence of the model's first hallucination.


"What a great problem! Let's dive into it step by step.
Step 1: Understand the problem We know that a store sold 235 robots in 12 months, and every month, they sold either 16, 20, or 25 robots. Our goal is to find the number of months in which exactly 20 robots were sold.
Step 2: Identify the possibilities Let's think about the possible combinations of robots sold each month:
16 robots/month: 12 months x 16 robots/month = 192 robots (short of 235)
20 robots/month: 12 months x 20 robots/month = 240 robots (more than 235)
25 robots/month: 12 months x 25 robots/month = 300 robots (way more than 235)
Since we need to reach 235 robots, we know that we can't have all months with the same number of robots sold. We need to mix and match the possibilities.
Step 3: Create a system of equations Let's introduce variables to represent the number of months for each type of robot sale:
Let x be the number of months with 16 robots sold.
Let y be the number of months with 20 robots sold.
Let z be the number of months with 25 robots sold.
We know that the total number of months is 12, so we can write an equation:
x + y + z = 12 ... (Equation 1)
We also know that the total number of robots sold is 235. We can set up another equation:
16x + 20y + 25z = 235 ... (Equation 2)
Step 4: Solve the system of equations We can solve Equation 1 for x: x = 12 - y - z
Substitute this expression for x into Equation 2:
16(12 - y - z) + 20y + 25z = 235
Expand and simplify:
192 - 16y - 16z + 20y + 25z = 235
Combine like terms:
4y + 9z = 43
Now, we need to find integer solutions for y and z. Since y and z represent months, they must be whole numbers.
Step 5: Find the solution After some trial and error, we find a solution:
y = 3 (months with 20 robots sold) z = 3 (months with 25 robots sold)
Substitute these values into Equation 1 to find x:
x + 3 + 3 = 12 => x = 6 (months with 16 robots sold)
Step 6: Verify the solution Let's calculate the total number of robots sold:
6 months x 16 robots/month = 96 robots
3 months x 20 robots/month = 60 robots
3 months x 25 robots/month = 75 robots
Total robots = 96 + 60 + 75 = 235
The solution checks out!
Answer: The store sold exactly 20 robots in 3 months."

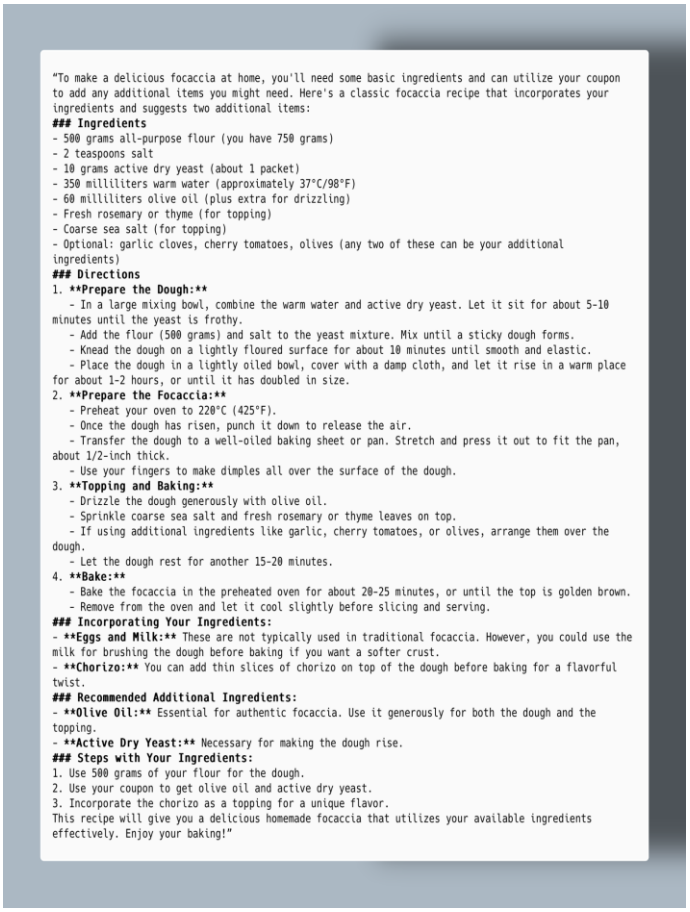
Fig. 5. Llama3 response.

B. *Creating a Culinary Recipe, Having a List of Ingredients Available*

Request 2: "I have the following ingredients: 500 milliliters of milk, six eggs, 750 grams of flour, 250 grams of chorizo and a coupon for two additional ingredients (it can be whatever I decide in whatever quantity). I want to make a focaccia but I

don't know which are the exact ingredients and the quantities for that. Please tell me a good recipe of focaccia to make in house."

ChatGPT Response**,** like in Fig. 6.



"To make a delicious focaccia at home, you'll need some basic ingredients and can utilize your coupon to add any additional items you might need. Here's a classic focaccia recipe that incorporates your ingredients and suggests two additional items:

Ingredients

- 500 grams all-purpose flour (you have 750 grams)
- 2 teaspoons salt
- 10 grams active dry yeast (about 1 packet)
- 350 milliliters warm water (approximately 37°C/98°F)
- 60 milliliters olive oil (plus extra for drizzling)
- Fresh rosemary or thyme (for topping)
- Coarse sea salt (for topping)
- Optional: garlic cloves, cherry tomatoes, olives (any two of these can be your additional ingredients)

Directions

1. ****Prepare the Dough:****
   - In a large mixing bowl, combine the warm water and active dry yeast. Let it sit for about 5-10 minutes until the yeast is frothy.
   - Add the flour (500 grams) and salt to the yeast mixture. Mix until a sticky dough forms.
   - Knead the dough on a lightly floured surface for about 10 minutes until smooth and elastic.
   - Place the dough in a lightly oiled bowl, cover with a damp cloth, and let it rise in a warm place for about 1-2 hours, or until it has doubled in size.
2. ****Prepare the Focaccia:****
   - Preheat your oven to 220°C (425°F).
   - Once the dough has risen, punch it down to release the air.
   - Transfer the dough to a well-oiled baking sheet or pan. Stretch and press it out to fit the pan, about 1/2-inch thick.
   - Use your fingers to make dimples all over the surface of the dough.
3. ****Topping and Baking:****
   - Drizzle the dough generously with olive oil.
   - Sprinkle coarse sea salt and fresh rosemary or thyme leaves on top.
   - If using additional ingredients like garlic, cherry tomatoes, or olives, arrange them over the dough.
   - Let the dough rest for another 15-20 minutes.
4. ****Bake:****
   - Bake the focaccia in the preheated oven for about 20-25 minutes, or until the top is golden brown.
   - Remove from the oven and let it cool slightly before slicing and serving.

Incorporating Your Ingredients:

- ****Eggs and Milk:**** These are not typically used in traditional focaccia. However, you could use the milk for brushing the dough before baking if you want a softer crust.
- ****Chorizo:**** You can add thin slices of chorizo on top of the dough before baking for a flavorful twist.

Recommended Additional Ingredients:

- ****Olive Oil:**** Essential for authentic focaccia. Use it generously for both the dough and the topping.
- ****Active Dry Yeast:**** Necessary for making the dough rise.

Steps with Your Ingredients:

1. Use 500 grams of your flour for the dough.
2. Use your coupon to get olive oil and active dry yeast.
3. Incorporate the chorizo as a topping for a unique flavor.

This recipe will give you a delicious homemade focaccia that utilizes your available ingredients effectively. Enjoy your baking!"

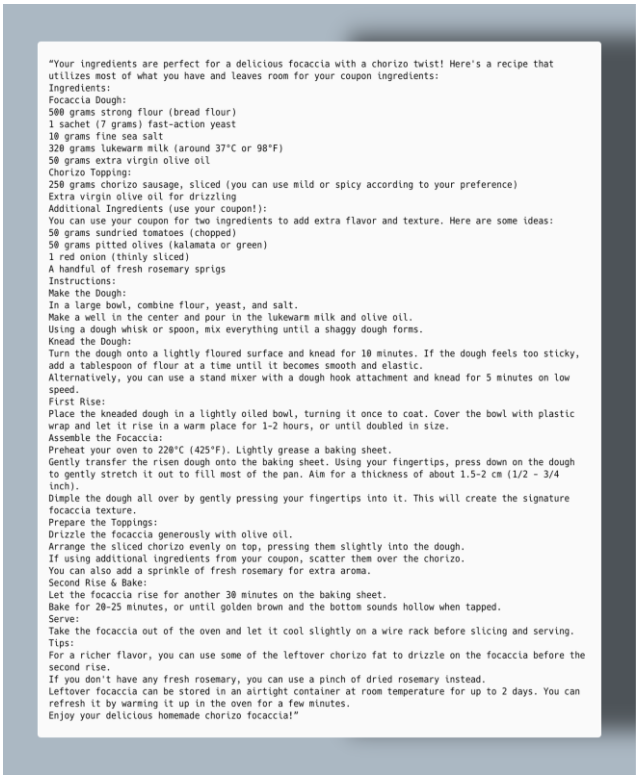
Fig. 6. ChatGPT response to request2.

Rating from ChatGPT – This is by far the best, most comprehensive and detailed answer ChatGPT has provided yet. Not only did he describe in detail every step for preparing and cooking the product, but he also made a number of correct suggestions regarding the coupon mentioned in the request: how I can use it and what missing ingredients from the list I can purchase with it.

Then, he mentioned that only 500g of the 750g of flour should be used for the proposed recipe, and he also realized that from the list of ingredients provided, two of them are not used for the focaccia recipe. Awesome, ChatGPT!

Gemini Response, like in Fig. 7.

Rating from Gemini – Gemini, this time had a good performance. The answer provided is correct, well structured, explained in detail. The ingredients used are the correct ones for a focaccia, as well as the method of preparation and cooking of the product. He also gave some tips at the end on how to improve the recipe, not bad at all, but not up to ChatGPT's level. Also, he didn't exactly understand the use of the coupon, the idea of the request was to purchase ingredients missing from the list with its help. However, on the whole it performed well the Gemini model, which until now has accustomed us to not very accurate answers.



"Your ingredients are perfect for a delicious focaccia with a chorizo twist! Here's a recipe that utilizes most of what you have and leaves room for your coupon ingredients:

Ingredients:

Focaccia Dough:

500 grams strong flour (bread flour)
1 sachet (7 grams) fast-action yeast
10 grams fine sea salt
320 grams lukewarm milk (around 37°C or 98°F)
50 grams extra virgin olive oil

Chorizo Topping:

250 grams chorizo sausage, sliced (you can use mild or spicy according to your preference)
Extra virgin olive oil for drizzling

Additional Ingredients (use your coupon!):

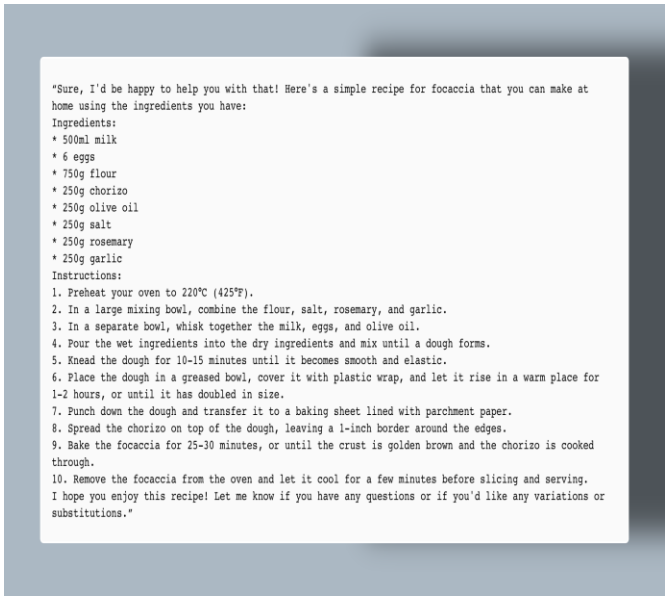
You can use your coupon for two ingredients to add extra flavor and texture. Here are some ideas:
50 grams sundried tomatoes (chopped)
50 grams pitted olives (kalamata or green)
1 red onion (thinly sliced)
A handful of fresh rosemary sprigs

Instructions:

Make the Dough:
In a large bowl, combine flour, yeast, and salt.
Make a well in the center and pour in the lukewarm milk and olive oil.
Using a dough whisk or spoon, mix everything until a shaggy dough forms.
Knead the Dough:
Turn the dough onto a lightly floured surface and knead for 10 minutes. If the dough feels too sticky, add a tablespoon of flour at a time until it becomes smooth and elastic.
Alternatively, you can use a stand mixer with a dough hook attachment and knead for 5 minutes on low speed.
First Rise:
Place the kneaded dough in a lightly oiled bowl, turning it once to coat. Cover the bowl with plastic wrap and let it rise in a warm place for 1-2 hours, or until doubled in size.
Assemble the Focaccia:
Preheat your oven to 220°C (425°F). Lightly grease a baking sheet.
Gently transfer the risen dough onto the baking sheet. Using your fingertips, press down on the dough to gently stretch it out to fill most of the pan. Aim for a thickness of about 1.5-2 cm (1/2 - 3/4 inch).
Dimple the dough all over by gently pressing your fingertips into it. This will create the signature focaccia texture.
Prepare the Toppings:
Drizzle the focaccia generously with olive oil.
Arrange the sliced chorizo evenly on top, pressing them slightly into the dough.
If using additional ingredients from your coupon, scatter them over the chorizo.
You can also add a sprinkle of fresh rosemary for extra aroma.
Second Rise & Bake:
Let the focaccia rise for another 30 minutes on the baking sheet.
Bake for 20-25 minutes, or until golden brown and the bottom sounds hollow when tapped.
Serve:
Take the focaccia out of the oven and let it cool slightly on a wire rack before slicing and serving.
Tips:
For a richer flavor, you can use some of the leftover chorizo fat to drizzle on the focaccia before the second rise.
If you don't have any fresh rosemary, you can use a pinch of dried rosemary instead.
Leftover focaccia can be stored in an airtight container at room temperature for up to 2 days. You can refresh it by warming it up in the oven for a few minutes.
Enjoy your delicious homemade chorizo focaccia!"

Fig. 7. Gemini response to request2.

Llama3 Response**,** like in Fig. 8.



"Sure, I'd be happy to help you with that! Here's a simple recipe for focaccia that you can make at home using the ingredients you have:

Ingredients:

* 500ml milk
* 6 eggs
* 750g flour
* 250g chorizo
* 250g olive oil
* 250g salt
* 250g rosemary
* 250g garlic

Instructions:

1. Preheat your oven to 220°C (425°F).
2. In a large mixing bowl, combine the flour, salt, rosemary, and garlic.
3. In a separate bowl, whisk together the milk, eggs, and olive oil.
4. Pour the wet ingredients into the dry ingredients and mix until a dough forms.
5. Knead the dough for 10-15 minutes until it becomes smooth and elastic.
6. Place the dough in a greased bowl, cover it with plastic wrap, and let it rise in a warm place for 1-2 hours, or until it has doubled in size.
7. Punch down the dough and transfer it to a baking sheet lined with parchment paper.
8. Spread the chorizo on top of the dough, leaving a 1-inch border around the edges.
9. Bake the focaccia for 25-30 minutes, or until the crust is golden brown and the chorizo is cooked through.
10. Remove the focaccia from the oven and let it cool for a few minutes before slicing and serving.

I hope you enjoy this recipe! Let me know if you have any questions or if you'd like any variations or substitutions."

Fig. 8. Llama3 response to request2.

Rating from Llama3 – The Llama3 model used all the ingredients on the list plus some added, and invented a recipe. We could say that he was creative, but comparing the answer with that of the ChatGPT model or even Gemini, I think more that he did not understand exactly the requirement. He did not

mention anything about the coupon, and he also used two ingredients that are not part of the focaccia recipe, ingredients that, at least ChatGPT specified, should not be used. It cannot be compared on this criterion with the first two models, once again remaining in last place. From what I have analyzed so far, it seems that the Llama 3 is not a stable model.

V. DISCUSSIONS

Following the comparative analysis carried out in this paper, we have learned some important aspects about these large linguistic patterns. First, these models are not stable. Although it is true that they were trained on massive data sets, these models have difficulties in solving more complex problems, logic, etc. Also, another argument to support this conclusion is the fact that their answers differ from one request to another, so much so that the final result can be completely different, and here I refer in particular to the Llama3 model. Of the 3 models chosen for comparative analysis, Gemini and Llama 3 are the most unstable.

An important aspect that deserves to be mentioned and put into practice is the observance of the principles of prompt engineering. An essential principle is to assess the quality of models when they do not provide a correct or true answer. Thus, these large language models can improve their performance and accuracy in future responses. This is confirmed by the analysis where ChatGPT's response improved after giving feedback that it did not respect the text size range specified in the request, ultimately providing the correct and complete response. The same can be said about the Gemini model, which improved its response after receiving feedback and respected the text size range specified in the request. Another principle of agile engineering that is very important to follow is providing direction. When interacting with a large language model, it is crucial to provide as much context as possible and to be as specific as possible about what we want to get from that model.

Regarding the performance and accuracy of the three models, observing the table. We can immediately conclude that ChatGPT is the language model that performed best in most of the criteria if we take into account the fact that after evaluating the quality of the answer and giving the feedback, it gave the correct final answer in case of the first evaluation criterion. Also, the ChatGPT language model provided accuracy and precision in most responses.

The Gemini model performed poorly overall, hallucinating several times and not giving accurate answers. These things are supported by several examples, namely: failure to respect the specified size range although we evaluated the quality and provided feedback on this aspect, it is not able to solve a math problem that also involves logic, as well as it is not even able to generate a code in the Python language that does not return an error and works correctly, and in the case of the interview simulator it performed very poorly, compared to the other two big language models, ChatGPT and Llama 3. A positive aspect in the Gemini model is the fact that it does not have an extremely important limitation, namely harmful or toxic results. Gemini, led by Llama 3, demonstrated that they are not limited from this point of view. We can't say the same about ChatGPT, unfortunately.

In terms of performance and accuracy of the answers provided, Llama 3 model is in the 2nd place, so it is located in the middle of the ranking. The worst hallucination from my point of view of this model was in the case of criterion 1 given by the wrong answer to the math problem and the incorrect focaccia recipe.

In conclusion, these large language models are far from stable, far from hallucinating, and even far from having a human understanding of a task. Although a lot is invested in this field, and many articles describe these models as being very capable, in essence, there is still a lot of work to be done before creating a machine with "human" characteristics. It is possible that in 10 years these models will have excellent performance and accuracy to match, but at the moment, although such a model can give you a very elaborate answer, most of the time, it is not useful in everyday reality, as we saw in the interview simulator. It is true, however, that such a model is really helpful, it can give you answers to guide you in your tasks if you know how to ask for these answers.

VI. CONCLUSIONS

Each model has its strengths and is designed to excel in different areas. ChatGPT (GPT-4) is versatile and strong in general-purpose conversational AI. Gemini leverages Google's extensive data and infrastructure for advanced contextual understanding and multilingual capabilities. Llama3 focuses on efficiency and performance, making it suitable for applications requiring lightweight and resource-efficient solutions. The choice between them would depend on the specific requirements of the application, such as accuracy, computational resources, and the need for real-time data integration.

Nowadays, these large language models are massively used in big companies and corporations. By incorporating these models in various fields, companies have witnessed improved customer interactions, improved content generation and efficient data analysis. The large language models have emerged as a transformative force in natural language processing, reshaping the way industries interact with and use text data.

As future directions, we can affirm that the insights gained from this comparative analysis highlight the importance of selecting the appropriate language model based on specific application needs and resource constraints. Future developments in large language models should focus first on improving accuracy and reducing bias where efforts should be made to enhance the accuracy of responses and minimize biases, particularly for models trained on diverse datasets; second on enhancing resource efficiency, continued innovation in model architecture improvement to optimize performance while reducing computational requirements that will be crucial, and third expanding accessibility and use cases that can increasing the accessibility of these models and broadening their applicability across different domains and industries that will drive further advancements in artificial intelligence.

In conclusion, ChatGPT, Gemini, and Llama3 each offer unique benefits tailored to different applications, and their continued evolution will play a significant role in the advancement of natural language processing technologies.

REFERENCES

[1] J. Phoenix and M Taylor, “Prompt engineering of generative AI, future-proof inputs of reliable AI outputs, ”in O'Reilly Media, vol. 1, 422 pag, 2024.

[2] Toloka. (n.d.), "History of LLMs". 09 May2024. [Online]. Available: https://toloka.ai/blog/history-of-llms/ [Accessed 18 June 2024].

[3] Wagh, A. “Open AI: Understand foundational concepts of ChatGPT and cool stuff you can explore“. 2 April 2023. Medium. [Online]. Available: https://medium.com/@amol-wagh/open-ai-understand-foundational-concepts-of-chatgpt-and-cool-stuff-you-can-explore-a7a77baf0ee3 [Accessed 18 June 2024].

[4] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., & Polosukhin, I, “All attention you need“, In Advances in Neural Information Processing Systems, pp. 5998-6008, 2017.

[5] ChatGPT. (2024). [Online]. Available: https://chatgpt.com/ [Accessed 18 March 2024].

[6] Google Gemini. (2024). [Online]. Available: https://gemini.google.com/app [Accessed 20 March 2024].

[7] Hugging Face. (2024). Llama 3 chatbot. [Online]. Available: https://huggingface.co/spaces/Be-Bo/llama-3-chatbot_70b [Accessed 28 March 2024].

[8] ChatGPT Guide. (2024, May 2). What is normalization? LLMs explained. [Online]. Available: https://www.chatgptguide.ai/2024/03/02/what-is-normalization-llms-explained/#:~:text=of%20the%20dataset.-,Normalization%20in%20LLMs,process%20and%20understand%20the%20data [Accessed 28 May 2024].

[9] Run:AI. (2024). A guide to large language model (LLM) training. [Online]. Available: https://www.run.ai/guides/machine-learning-engineering/llm-training [Accessed 30 May 2024].

[10] Shaip. (2024). A guide to large language model (LLM). [Online]. Available: https://ro.shaip.com/blog/a-guide-large-language-model-llm/ [Accessed 01 June 2024].