Lie Detection - Prompting Llama 2 on three datasets: Intentions, Opinions and Memories

Furkan Can Algan (ID: 2085308)
Camilo Betancourt Nieto (ID: 2087189)
Cameron Kelahan (ID: 2071947)
Joan Orellana Rios (ID: 2081188)
Cristian Granchelli (ID: 2071977)

February 2024

1 Introduction

Artificial intelligence tools are becoming increasingly popular, especially those powered by large language models (LLMs). These models are created to understand and generate human language. Some researchers suggest that LLMs primarily recognize patterns and generate sequences based on probabilistic patterns observed during training, without a true understanding of meaning. On the other hand, there's an argument that, as the model's scale increases, these tools could potentially reach a level of intelligence comparable to humans. Recent studies show that larger LLMs, with more training data and computational power, exhibit more complex behaviors, indicating a deeper grasp of language and context. [19]

Current research suggests that we are in a transitional phase, and there is a possibility that large language models (LLMs) may reach human-level intelligence in the future. However, as of now, even though LLMs perform well at language-related tasks, they still have some limitations and are not considered to be equivalent to human intelligence. It is advisable not to rely on them for making judgments [12], particularly in tasks requiring causal reasoning and planning, such as understanding consequences or predicting future outcomes. LLMs may also reflect biases and errors present in their training data. Given these constraints, it is crucial to approach LLMs with caution and a critical perspective. Further studies are necessary to evaluate their potential to reach human-level intelligence and the potential applications that may arise from these improvements.

In this context, lie detection is one of the tasks where the relevance of discussing the intelligence of large language models (LLMs) becomes evident when considering their capacity to analyze deceptive information in speech and text. Therefore, if LLMs were to be used for lie detection purposes, it is crucial to be mindful of their existing limitations, so trying to identify and understand them becomes a necessity. This paper explores three different approaches to measuring the ability of the Llama 2 LLM to detect truth versus deception.

1.1 Lie detection

As online shopping has become increasingly popular, we often buy things and plan activities such as special occasions or holidays based on online reviews. However, it's important to note that these reviews are susceptible to manipulation. Therefore, the detection of deceptive information is crucial as it directly impacts our day-to-day experiences. [3].

Previous research on deceit detection revealed lack of effectiveness of humans, since they obtained accuracy rates just above chance levels [2]. Additionally, evidence suggests that professionals, including psychologists, detectives, and judges, do not exhibit greater accuracy than students and other citizens in this task [1].

In this context, there is increasing interest in automatically detecting deception through language analysis [7, 6]. For this reason, understanding the behavior of LLMs is a subject that is worth studying.

With this in mind, we will use Llama 2, a large language model developed by Meta, to perform lie detection tests using three datasets that focus on different aspects: opinions, memories and intentions. First, we provide a brief discussion on prompt engineering, since it is a key element of the procedures we will employ.

2 Prompt Engineering

Prompt engineering involves designing effective and precise prompts or instructions to guide language models to generate desired outputs. It is an important aspect of using these models to achieve specific tasks. The goal is to construct prompts that elicit accurate and contextually relevant responses from the model. Engineers and users experiment with different prompt structures, styles, and strategies to optimize the performance of language models for diverse applications such as translation, summarization, question answering, and more. Prompt engineering helps fine-tune the behavior of the model and tailor its responses to meet the user's requirements effectively.

There are various types of prompting techniques employed in working with language models. Some common types include:

- Zero-shot Prompting
- Few-shot Prompting
- Chain of Thought Prompting
- Self-Consistency
- Tree Of Thoughts

2.1 Zero-shot Prompting

Zero-shot prompting is a technique where the model is prompted to generate responses without being fine-tuned on specific tasks or domains. This means that the model is expected to understand and respond to prompts across various topics, even if it has not been explicitly trained on those specific prompts. It leverages the general knowledge and understanding encoded within the pre-trained model to generate relevant responses to a wide range of prompts, making it a versatile and efficient method for interacting with language models across different domains without the need for extensive fine-tuning. [5]

Example:

Prompt:

```
Classify the text into neutral, negative, or positive.
Text:  I think the vacation is okay.
Sentiment:
```

Output:

```
Neutral
```

2.2 Few-shot Prompting

Few-shot prompting is a technique where the model is prompted to generate responses after being fine-tuned on a small amount of task-specific examples or data. Unlike zero-shot prompting, which does not involve task-specific fine-tuning, few-shot prompting allows the model to adapt to specific tasks or domains by providing a limited amount of task-specific examples.

Example:

Prompt:

```
A "whatpu" is a small, furry animal native to Tanzania.  An example of a sentence
that uses the word whatpu is:
We were traveling in Africa and we saw these very cute whatpus.
To do a "farduddle" means to jump up and down really fast.  An example of a sentence
that uses the word farduddle is:
```

Output:

```
When we won the game, we all started to farduddle in celebration.
```

2.3 Chain of Thought Prompting

Chain of thought prompting involves connecting multiple prompts together to create a more interactive and dynamic conversation with a language model. This technique allows for multi-turn dialogues and more detailed responses by linking prompts in a sequence. By chaining prompts, the conversation can evolve in a structured manner, enabling the language model to generate responses that build upon previous inputs. This approach facilitates a more coherent and contextually relevant interaction, guiding the model to produce informative and connected outputs based on the progression of prompts.

Example:

Prompt:

```
The odd numbers in this group add up to an even number:  4, 8, 9, 15, 12, 2, 1.
A: Adding all the odd numbers (9, 15, 1) gives 25.  The answer is False.
The odd numbers in this group add up to an even number:  17, 10, 19, 4, 8, 12, 24.
A: Adding all the odd numbers (17, 19) gives 36.  The answer is True.
The odd numbers in this group add up to an even number:  16, 11, 14, 4, 8, 13, 24.
A: Adding all the odd numbers (11, 13) gives 24.  The answer is True.
The odd numbers in this group add up to an even number:  17, 9, 10, 12, 13, 4, 2.
A: Adding all the odd numbers (17, 9, 13) gives 39.  The answer is False.
The odd numbers in this group add up to an even number:  15, 32, 5, 13, 82, 7, 1.

A:
```

Output:

```
Adding all the odd numbers (15, 5, 13, 7, 1) gives 41.  The answer is False.
```

In recent years, "Zero-Shot Chain of Thought" has also been explored and suggested as a legitimate prompt engineering approach. In their 2022 paper, Kojima et al. reported improvement in zero-shot prompting by simply adding "Let's think step by step" at the end of the original zero-shot prompt [10].

2.4 Self-Consistency

Self-consistency involves generating multiple diverse reasoning paths using few-shot CoT and then selecting the most consistent answer from these generations. This approach significantly enhances the performance of CoT prompting, particularly in tasks related to arithmetic and common-sense reasoning.

2.5 Tree Of Thoughts

Tree of Thoughts (ToT) is a technique that extends the idea of chain-of-thought prompting. It advocates for the exploration of intermediate steps or thoughts as a means for solving complex problems using language models. This approach emphasizes the importance of navigating through a series of logical steps or considerations to enhance the problem-solving capabilities of AI-driven language models.

ToT maintains a structure called tree of thoughts, in which each thought represent a coherent language sequence that acts as a step towards problem resolution. This method allows an LLM to assess how each thought contributes to problem-solving via a structured reasoning pathway.

3 Lie detection with the Deceptive Opinions dataset

In this case, we worked with the Deceptive Opinions (DecOp) corpus [3], a multilingual and multi-domain dataset developed for automatic deceit detection tasks. The DecOp dataset provides a ground to tackle two of the unsolved issues in the automatic detection of deception: cross-domain and cross-language classification tasks.

Regarding the cross-domain issue, in the paper in which the DecOp corpus was introduced, the authors mention that, although the generalization of classification performances when dealing with data coming from different domains has been previously studied [8][16][14], there are few large labelled datasets in the scientific literature that allow to do these type of studies, which motivates the creation of the DecOp corpus.

Furthermore, with respect to the cross-language matter, the authors argue that most of the studies of automatic deceit deception have mainly concentrated on English. Meanwhile, a small number of works have explored the "English-based linguistic clues of deception in predicting deceit in languages other than English" [3]. According to the authors, since the need for further research in this topic has been supported before [18][13][20], and there is also scarcity of data sources that allow to perform these type of studies, the DecOp corpus aims to address this situation as well.

3.1 Structure of the dataset

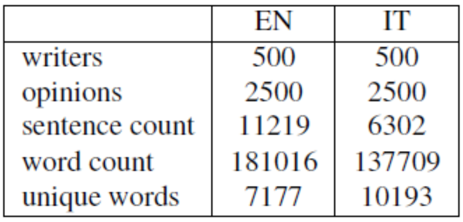
Considering the context provided, the DecOp corpus is a collection of first-person opinions from 1,000 people about five different topics: Abortion (A), Cannabis Legalization (CL), Euthanasia (E), Gay Marriage (GM), and Policy on Migrants (PoM). The sample of people was divided in 500 people from the US and 500 people from Italy. All participants used their native language to answer in a free text response modality.

The data collection procedure was designed so that half of the opinions for each topic and each language are truthful (they represent the person's actual opinion on that topic). The other half of the opinions are deceptive, meaning that they do not represent the person's true opinion but are written with the purpose of convincing a hypothetical reader that it is indeed their real point of view about the topic.

Also, it is worth mentioning that the dataset was split into training and test sets. The split was done in a balanced way with respect to the ground truth. It assigned the responses of 400 writers to the training set, while the responses from the remaining 100 writers were assigned to the test set.

Thus, the dataset contains multilingual and multi-domain information that allows to perform cross-domain and cross-language investigations on deceit detection. The corpus also allows to carry out within-domain and author-based classification tasks, although this will not the main focus of this work.

Some examples of the opinions collected are shown in the original DecOp paper, while the following figure shows the structure of the data [3]:



	EN	IT
writers	500	500
opinions	2500	2500
sentence count	11219	6302
word count	181016	137709
unique words	7177	10193

Figure 1: DecOp statistics

3.2 Human performance on the DecOp dataset

The paper that introduced the DecOp corpus also described the results of a human performance experiment on the DecOp. This experiment was carried out to check the DecOp capabilities and provide a baseline for further analyses such as the one conducted in this project.

The experiment conducted was a binary classification task, where 100 participants were asked to categorize opinions in a cross-topic setting (so all domains were included). Each participant was asked to classify 10 opinions as truthful (labeled as "T") or deceptive (labeled as "F") in a binary modality. The opinions were randomly extracted from the test-set and the overall classification was computed separately for the US and the Italian participants.

The overall accuracy was 57.9% and 58% for the Italian and the US participants, respectively. These results confirmed the lack of efficiency of humans in detecting deception [3].

Additionally, the outcome of the experiment revealed the presence of the so-called truth-bias, a common human judgmental bias that affects people's abilities to detect deception [3]. The truth-bias could be defined as "the propensity of judging more often messages as truthful than deceptive, regardless of their actual veracity" [11]. The authors of the DecOp paper identified the presence of this cognitive bias based on an unbalanced tendency in both groups to classify opinions as truthful. Italian participants classified 79.3% of the opinions as truthful, whereas this proportion was 62.9% for the US participants.

3.3 Methodology for testing Llama 2 on the DecOp dataset

In this instance, the goal is to compare the behavior of the Llama 2 tool with respect to the human performance on the DecOp corpus. Also, instead of just replicating the experiment made with humans, we wanted to do some extra tests. For these reasons, given the nature of the data and the way that the human tests were performed, we used zero-shot and few-shots approaches as prompting techniques to assess Llama 2's performance on the dataset.

More specifically, we performed four types of tests for each language:

- Zero-shot tests.
- Few-shot test with two examples in the test opinion's language.
- Few-shots test with mixed-language examples.
- Few-shot test with two examples in a different language from the test opinion.

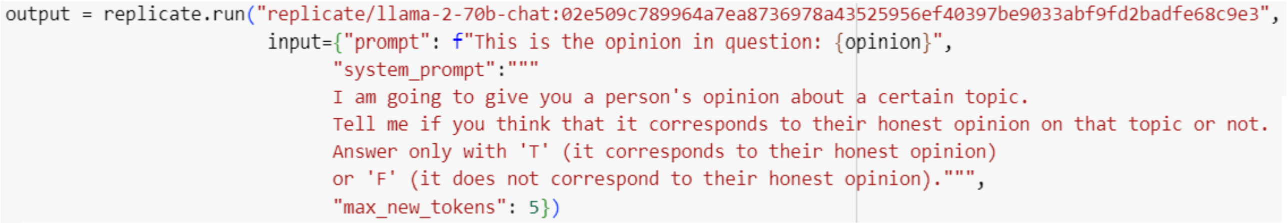
Next, we will provide a detailed description each methodologies.

3.3.1 Zero-shot tests

This approach aims to imitate the test made on humans as closely as possible. First, we took 10 randomly-selected balanced subsets of 10 opinions from the whole test set and asked Llama 2 to classify them as truthful or deceptive. We did this for both languages. This way, we could try to mimic what was done with the 100 human testing participants.

We used 10 subsets instead of the 100 of the original test because of computational reasons. In this sense, it is worth mentioning that, for this dataset, we used the 70B version of Llama 2 with the *Replicate* API for Python (it requires more processing resources than other versions and *Replicate* imposes a limit on free credits). Also, we chose to do it like this instead of just picking 100 random opinions and passing them to Llama 2 because, although it is not explicitly stated in the DecOp paper, by randomly giving 10 opinions to each human participant to classify, the sampling approach may admit repetition of elements among samples. Even though it might be a minor detail, our methodology allows to replicate this phenomenon.

We used a zero-shot prompting technique, so we did not give any examples to the model prior to the classification instruction. In Figure 2, there is an example of the prompts given to Llama 2 for this test.



```
output = replicate.run("replicate/llama-2-70b-chat:02e509c789964a7ea8736978a43525956ef40397be9033abf9fd2badfe68c9e3",
                       input={"prompt": f"This is the opinion in question: {opinion}",
                              "system_prompt":"""
                              I am going to give you a person's opinion about a certain topic.
                              Tell me if you think that it corresponds to their honest opinion on that topic or not.
                              Answer only with 'T' (it corresponds to their honest opinion)
                              or 'F' (it does not correspond to their honest opinion).""",
                              "max_new_tokens": 5})
```

Figure 2: Prompt for the zero-shot test

Given that the experiment conducted on humans had a binary response (it could be either "T" or "F"), we asked Llama 2 to answer in that same way by including that instruction in the system prompt (a text that is included before the actual prompt to help guide or constrain the model behavior). Additionally, we limited the maximum number of tokens to 5 (in the API's website it is stated that a word is generally 2-3 tokens [17]), just to avoid long answers in our output. As for the temperature parameter, which controls the randomness of the output (a value greater than 1 is random and 0 is deterministic), we used the default setting: 0.75.

3.3.2 Few-shot test with two examples in the test opinion's language

In this case, we used a few-shot prompting technique, where we provided two examples to the model before asking it to classify the test opinion. To be more precise, we passed two opinions with their corresponding label.

What is distinctive in this test in comparison to the next two is that both of the example opinions were written in the same language as the test opinion. This way, we are checking how well the model can pick up on deception cues within a specific language and restricted context.

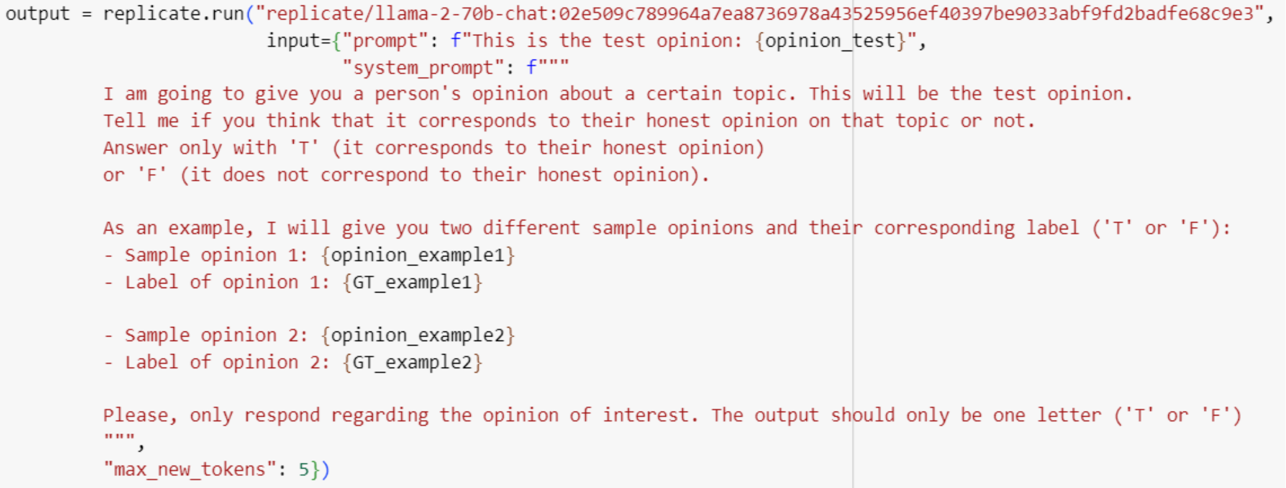
We selected the example opinions by randomly picking two observations of the corresponding training set (the one in English or the one in Italian, depending on the language of the test opinion) for each of the elements in the subsets that we obtained for testing in the previous section. Therefore, the sampled opinions could be either truthful or deceptive. Also, the approach we used is cross-topic, so the examples picked could be about any of the domains. Below, there is an example of the opinions given to Llama 2 prior to the classification of a test opinion in English:

```
   Example opinion 1:  "i think that it can be moral.  If a person has lived
passed their usefulness then why shouldn't we consider it?  more so if the
person themselves are the ones asking for it.  they wish to not be a burden to
anyone so why is it okay for us to keep them from their wishes."
Label of opinion 1:  F

   Example opinion 2:  "Cannabis should not be legalized for recreational use.
Although there are legitimate uses for cannabis medicinally, there are unknowns
for the effects of using it recreationally.  Not enough studies have been
conducted, and there are new studies being released that show troubling results
from marijuana use, such as infertility, cognitive effects, and structural
changes in the brain for teens.  By legalizing it, more young people will gain
access to it and use it because they see it as harmless."
Label of opinion 2:  T
```

We tried some different formats to pass as the system prompt and the prompt for this test. In the end, the one that worked best (in the sense that the model complied with giving us just the label of the test opinion instead of long answers) was the following one:

The rest of the parameters were left as in the zero-shot test.



```
output = replicate.run("replicate/llama-2-70b-chat:02e509c789964a7ea8736978a43525956ef40397be9033abf9fd2badfe68c9e3",
                        input={"prompt": f"This is the test opinion: {opinion_test}",
                               "system_prompt": f"""
         I am going to give you a person's opinion about a certain topic. This will be the test opinion.
         Tell me if you think that it corresponds to their honest opinion on that topic or not.
         Answer only with 'T' (it corresponds to their honest opinion)
         or 'F' (it does not correspond to their honest opinion).

         As an example, I will give you two different sample opinions and their corresponding label ('T' or 'F'):
         - Sample opinion 1: {opinion_example1}
         - Label of opinion 1: {GT_example1}

         - Sample opinion 2: {opinion_example2}
         - Label of opinion 2: {GT_example2}

         Please, only respond regarding the opinion of interest. The output should only be one letter ('T' or 'F')
         """,
         "max_new_tokens": 5})
```

Figure 3: Prompt for the few-shot tests

3.3.3 Few-shot test with mixed-language examples

This test is almost identical to the previous one. The only difference is the language of the examples passed to Llama 2 as context. In this instance, we passed one opinion in English and one opinion in Italian for each observation in the test subsets. As in the previous case, the example opinions were randomly selected from the training sets and could contain any of the five domains. The following ilustrates the types of examples passed as context for a test opinion in English (but the same would apply for Italian):

```
   Example opinion 1:  "I am of two minds of this issue, because it has become
so confusing recently.  I think better conditions should be put into place for
asylum seekers, and primarily to eliminate those who have real criminal records.
Otherwise, I think migrants should be allowed to come to the U.s.  in many cases."
Label of opinion 1:  T

   Example opinion 2:  "Si dice che legalizzare la cannabis contrasti la
criminalità organizzata che ha sopra di essa un monopolio.  Ma allora aboliamo
anche l'omicidio su commissione, visto che anche quello è una fonte di entrate
economiche per le mafie!  Inoltre, immaginate cosa voglia dire riempire le nostre
strade di drogati?  Ma stiamo scherzando?  E poi, oggi la cannabis, domani la
cocaina, dopodomani l'eroina."
Label of opinion 2:  F
```

3.3.4 Few-shot test with two examples in a different language from the test opinion

In this last experiment, we also provided two opinions as context before the classification. Again, the only difference with the previous tests is the language of these example opinions. This time, both examples were given in a different language from the one of the test opinion. An instance of the examples given to a test opinion in English would look as follows:

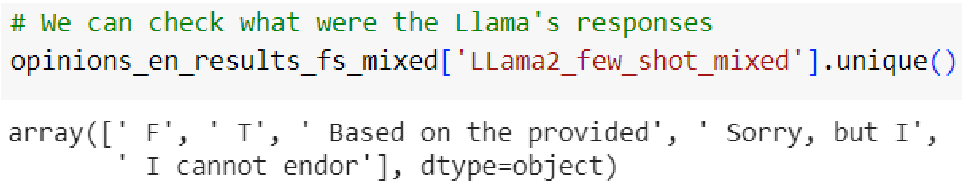
```
   Example opinion 1:  "Il concetto di nazione ha radici molto antiche, ma certamente
quella che la colora maggiormente é quella che ci rende tutti parte di uno stesso
```

```
confine, intersecati in qualcosa che viviamo sin dalle nostre origini.  Bisogna
difendere ciò che è nostro e non permettere che persone che non lo conoscano entrino
e abbiano diritti paritetici ai nostri in un contesto di cui non sanno niente ma
che è solo la loro ancora di salvataggio."
Label of opinion 1:  F

   Example opinion 2:  "Completamente favorevole al matrimonio tra due persone che
si amano e vogliono condividere la loro vita insieme, indipendentemente dal sesso."
Label of opinion 2:  T
```

3.4 Llama 2 results on the DecOp dataset

Most of the time, Llama 2 gave as an output the binary label ("T" or "F") as it was instructed. This way, computing the accuracy was straightforward. However for a few examples, the model yielded a different kind of response in which it tried to write a longer text such as the following (the strings are truncated because of the settings we used for the maximum number of tokens):

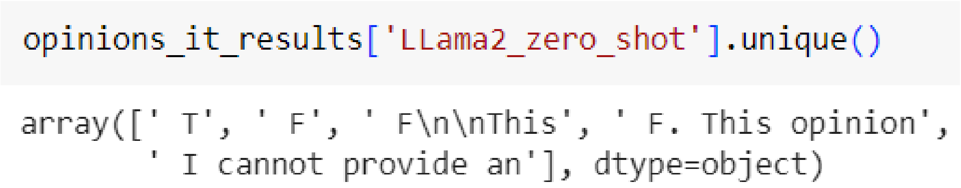


```
# We can check what were the Llama's responses
opinions_en_results_fs_mixed['LLama2_few_shot_mixed'].unique()
```

```
array([' F', ' T', ' Based on the provided', ' Sorry, but I',
       ' I cannot endor'], dtype=object)
```

Figure 4: Examples of Llama 2's outputs for the DecOp corpus

Since these responses do not align with the instruction, the classification was considered as incorrect in these cases. A special case that only occurred in the zero-shot tests with the Italian dataset, was that Llama 2 gave the label and then tried to give an explanation. In this cases, we manually adjusted the answer and compared the resulting response with the opinion's true label:



```
opinions_it_results['LLama2_zero_shot'].unique()
```

```
array([' T', ' F', ' F\n\nThis', ' F. This opinion',
       ' I cannot provide an'], dtype=object)
```

Figure 5: Special examples of Llama 2's outputs for the DecOp corpus

Once we collected the model's responses for every task, we were able to compute the overall accuracy for each of the tests:

First, what stands out is that the overall accuracy in all tests was higher than the accuracy showed by humans on the same dataset (57.9% for Italian and 58% for English). This might help to explain why there is a growing interest in automatic detection of deception. Also we can see that the overall accuracy was better for all the tests conducted on the English dataset in comparison to the Italian counterparts. This result could be explained by the fact that English is more widely spoken, which results in Llama 2 having more experience with English than with Italian.

Prompting technique	Accuracy	Opinions classified as truthful
Zero-shot EN	71%	64%
Zero-shot IT	69%	76%
Few-shot EN test using EN examples	74%	60%
Few-shot EN test using IT examples	73%	65%
Few-shot EN test using mixed examples	76%	66%
Few-shot IT test using IT examples	71%	73%
Few-shot IT test using EN examples	71%	64%
Few-shot IT test using mixed examples	73%	68%

Table 1: Prompting Techniques and Accuracy for the Opinions Dataset

Additionally, we also observe zero-shot tests obtained worse but not so different results from those obtained with the few-shot techniques. One possible explanation for this might be that Llama 2 struggled at making the most of the context provided in the system prompt due to its length and complexity. This is coherent with the fact that, especially for the few-shot tests, we had to try several approaches to writing the system prompt in order to make Llama 2 respond in the desired format. Additionally, what we obtained for both test languages is that the best few-shot accuracy was achieved when the model received mixed examples (one in English and one in Italian) as context.

On the other hand, regarding the truth-bias, we observe that all tests had a proportion of opinions classified as true above 60%. If we compare the zero-shot tests with the results obtained by humans in this aspect (79.3% for Italian and 62.9% for English), we can see a similar behavior, even though the truth-bias effect is slightly less prominent in the Llama 2 results for the Italian dataset.

4 Lie Detection with the Memories dataset

4.1 Study on the research article: Quantifying the narrative flow of imagined versus autobiographical stories

This article [19] explores the differences between autobiographical and imagined stories by introducing a metric called "sequentiality." This metric measures the narrative flow of events in stories by drawing probabilistic inferences from a large language model. The study gives insights into the cognitive processes of storytelling and moves away from traditional approaches to analyze narratives. The authors use LLMs to explore narrative theories and consider the influences of common schema and personal experiences on the stories that people tell. The paper also discusses the potential applications of the research findings in various domains such as memory, reasoning, cultural influences, and motivations for storytelling.

4.1.1 Sequentiality Metric

The sequentiality metric measures narrative flow based on probabilities of story sentences given by LLMs. It compares the likelihood of each story sentence under two generative models: a contextual model and a topic-driven model. Higher values of sequentiality for sentences suggest that the sentences follow the common expectations given the context of the evolving story and topic, whereas lower values suggest that sentences deviate more from expectation, given the preceding sequence of sentences. Essentially, it quantifies how much a story follows the expected or common narrative flow for a specific story topic (schematic knowledge) versus how grounded it is in experiential details (episodic memory).

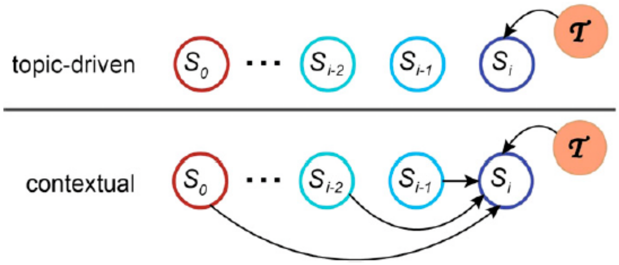


Figure 6: Graphical models depicting the two components of sequentiality

The image in Figure 6 shows the two components of sequentiality, illustrating the probabilistic relationship among consecutive sentences in a story about a specific topic. It illustrates the difference between the log-likelihood of a sentence conditioned only on the story topic (topic-driven model) and the log-likelihood of that sentence conditioned on both the story topic and all preceding sentences (contextual model), which are the key components of sequentiality.

4.1.2 Findings

The findings of the article highlight the variations in narrative flow between imagined and autobiographical stories. Specifically, the study reveals that retold autobiographical stories have a higher degree of sequentiality compared to fresh recollections. Moreover, autobiographical stories consist of a greater number of realis events terms and concrete words compared to imagined stories. Additionally, autobiographical stories contain a higher proportion of salient events in comparison to imagined stories. Interestingly, sentences with salient events exhibit lower sequentiality, indicating a deviation from the expected narrative flow.

Furthermore, the research searches through the proportion of salient events in a subset of stories, clarifying their significance in shaping the narrative content and emotional impact.

Moreover, the study investigates the relationships between event annotation, sequentiality, LIWC (Linguistic Inquiry and Word Count), and concreteness lexicons. This analysis provides a deeper understanding of how narrative flow, linguistic characteristics, and the presence of salient events intertwine, offering a perspective on the factors that contribute to the construction and coherence of narratives.

4.1.3 Dataset

The dataset for the study on narrative flow was collected from crowdworkers and consisted of thousands of diary-like stories. These stories were either about a recent remembered experience or an imagined story on the same topic. The data gathered included:

- **Autobiographical Stories**: Some of the crowdworkers were asked to write stories about remembered experiences, containing specific details drawn directly from personal experiences.
- **Imagined Stories**: Some of the crowdworkers were asked to write imagined stories, starting from a summary of a recolled story.
- **Retold Autobiographical Stories**: Crowdworkers who wrote the recalled story were asked to retell the same story after a period of time, ranging from 3 months to 6 months later.

The dataset was used to quantify the differences between autobiographical and imagined stories, analyze the narrative flow of events, and explore the influences of memory and reasoning on language generation processes.

4.2 Experiments

We used the 13b parameters Llama 2 version, setting the temperature parameter to 0.1, the repetition penalty parameter to 1.1 and the maximum number of tokens to 512. We accessed the model using the library Hugging Face.

The experiments have been run on a subset of the original dataset made of 100 examples.

This is how we implemented the prompt engineering techniques:

- **Zero-shots**: we asked the model to tell us if a story was recalled or imagined, with no context and without providing any examples. The prompt we used is:

  Can you tell if this is a imagined story or a recalled story?

  Story: "Here we inserted the story to evaluate"

  Answer with one word

  Answer:

- **Few-shots**: we asked the model to tell us if a story was recalled or imagined, giving it an example of an imagined story and an example of a recalled story. The prompt we used is:

  Question: Can you tell if this is a Imagined story or a Recalled story?

  Story: "Here we inserted the imagined story"

  Answer: Imagined

  Question: Can you tell if this is a Imagined story or a Recalled story?

  Story: "Here we inserted the recalled story"

  Answer: Recalled

  Question: Can you tell if this is a Imagined story or a Recalled story?

  Story: "Here we inserted the story to evaluate"

  Answer:

- **Chain of Thought with zero-shots**: we asked the model to tell us if a story was recalled or imagined, with no context and without providing any examples and asked the LLM to "think step by step". The prompt we used is:

  Question: Can you tell if this is a Imagined story or a Recalled story? Let's think step by step.

  Story: "Here we inserted the story to evaluate"

  Answer:

- **Chain of Thought with few-shots**: we asked the model to tell us if a story was recalled or imagined, giving it an example of an imagined story and an example of a recalled story and asked the LLM to "think step by step". The prompt we used is:

  Question: Can you tell if this is a Imagined story or a Recalled story? Let's think step by step.

  Story: "Here we inserted the imagined story"

  Answer: "Here we inserted the label with a justification"

  Question: Can you tell if this is a Imagined story or a Recalled story? Let's think step by step.

  Story: "Here we inserted the recalled story"

  Answer: "Here we inserted the label with a justification"

  Question: Can you tell if this is a Imagined story or a Recalled story? Let's think step by step.

  Story: "Here we inserted the story to evaluate"

  Answer:

- **Self-consistency**: we asked the model to tell us if a story was recalled or imagined, giving it an example of an imagined story and an example of a recalled story, for three times and then picked the majority class as answer. The prompt we used is:

  Question: Can you tell if this is a Imagined story or a Recalled story?

  Story: "Here we inserted the imagined story"

  Answer: "Here we inserted the label with a justification"

  Question: Can you tell if this is a Imagined story or a Recalled story?

  Story: "Here we inserted the label with a justification"

  Answer: Recalled

  Question: Can you tell if this is a Imagined story or a Recalled story?

  Story: "Here we inserted the story to evaluate"

  Answer:

- **Tree of Thoughts**: we asked the model to tell us if a story was recalled or imagined, with no context and without providing any examples, telling him to maintain a tree of thoughts. The prompt we used is:

  Imagine that three different experts have to say if a story is imagined or recalled. All experts will write down 1 step of their thinking, then share it with the group. After every expert shared its 1st step, they are going to repeat the process for multiple step. Please report all the steps. If any expert realises they are wrong at any point then they leave.

At the end of the answer you should say if the experts have chosen between Imagined and Recalled by answering Imagined if you think it is imagined or Recalled if you think it is recalled. They cannot say it is both Imagined and Recalled, they need to make a choice. At the end of the answer give me the final label by saying "Imagined" or "Recalled"

Question: Can you tell if this is a Imagined story or a Recalled story?

Story: "Here we inserted the story to evaluate"

Answer:

- **Ensamble**: we have implemented an ensamble method that uses a voting system, where each model votes for either "Imagined" or "Recalled" based on their output. The prediction is given by the class that obtains the highest number of votes.

4.3 Results

Prompting Technique	**Accuracy**
Zero-shot	51%
Few-shot	53%
Zero-shot Chain of Thought	59%
Few-shot Chain of Thought	54%
Self-Consistency	49%
Tree Of Thoughts	61%
Ensemble	58%

Table 2: Prompting Techniques and Accuracy for the Memories Dataset

- **Zero-shot**: Achieved an accuracy of 51%, which is slightly above random guessing (50% for a binary classification task). This suggests that without any prior examples or context, the LLM can barely distinguish between imagined and recalled stories.

- **Few-shot**: Improved slightly to 53% accuracy. Providing the model with a few examples helps a bit, but the increase is marginal, indicating that this task is challenging for the model even with a small amount of direct guidance.

- **Zero-shot Chain of Thought**: Shows a significant jump to 59% accuracy. This technique seems to enhance its performance, suggesting that the task benefits from explicit reasoning.

- **Few-shot Chain of Thought**: Has a slight decrease to 54%, surprisingly underperforming the zero-shot chain of thought. This might indicate that the additional examples might introduce some biases or overfitting, reducing the effectiveness of the reasoning process.

- **Self-Consistency**: Surprisingly drops to 49%, the lowest among the methods. This approach might not be effective for lie detection in this context, possibly due to the inherent ambiguity in distinguishing between imagined and recalled stories.

- **Tree Of Thoughts**: Achieves the highest accuracy at 61%. This advanced technique, likely involving a structured approach to reasoning with multiple branches of thought, appears most effective for this complex task. It suggests that breaking down the problem into smaller, manageable pieces and exploring various reasoning paths can significantly enhance performance.

- **Ensemble**: Shows 58% accuracy, which is robust but not as high as the Tree of Thoughts. Combining predictions from multiple models or techniques usually improves performance by leveraging their strengths and mitigating individual weaknesses. However, in this case, it does not outperform the best individual technique.

Humans, on average, have a performance accuracy of around 50% in lie detection tasks, which is essentially equivalent to random guessing in a binary classification scenario. This baseline suggests that without specific training or expertise, humans find it challenging to consistently identify lies, especially when the lies are about personal experiences or memories, which can be very subjective and nuanced.

Llama 2's performance on this dataset varies significantly across different prompting techniques. This indicates that under certain conditions and with specific methodologies, the LLM can surpass human performance in this task. The success of the Tree Of Thoughts approach, in particular, suggests that a structured and nuanced exploration of reasoning can provide an edge over the general human ability to detect lies, at least within the constraints of this dataset and task.

However, it is important to note that the LLM's performance, even at its best, is not overwhelmingly superior to human performance. The gains are modest, and the overall accuracy rates indicate that lie detection remains a challenging problem for both humans and machines. The effectiveness of the LLM depends heavily on the prompting technique used, highlighting the importance of methodological choices in leveraging AI capabilities.

Moreover, these results should be interpreted with caution. The performance of both humans and the LLM in this specific task might not generalize across different datasets.

5 Intentions

5.1 Overview of "How humans impair automated deception detection performance"

In this article [9], Kleinberg and Verschuere investigate the abilities of humans and machine learning algorithms to detect deceptive intentions, both separately and combined in 'hybrid' ways.

5.1.1 The Dataset

The dataset used in this study consists of statements collected from people through a web-interface. They were asked to "provide a statement about their most significant non-work-related activity in the next seven days" given the activity is "specific, has a clear start and end time, and should not be a continuous or daily activity." In two separate questions from the initial title of the activity, they were then asked to: "describe your activity as specifically as possible" and "what information can you give us to reassure us that you are telling the truth" [9].

Participants were randomly assigned the 'truthful' or 'deceptive' condition. If assigned 'truthful,' their given activity was genuinely going to occur. If assigned 'deceptive,' they were shown three examples of activities previously given by a 'truthful' participant and asked to identify if any of these randomly chosen events did not apply to them. If any, they were removed, and one of the remaining options was randomly chosen to be their assigned activity. The 'deceptive' participant was then asked to convincingly carry out the 2 supplemental questions.

After answering, participants were asked to describe their level of motivation with regards to providing a convincing response, and how certain they were they would carry out their actual activity (both on scales of 0-10). These values were not included in the dataset, but were used as a measurement of the quality of the responses. With an initial corpus of 2,027 statements, they removed answers that were too short (less than 15 words) and answers where the response to the second question closely resembled the response to the first question. In the end, the final corpus consisted of 1,640 activities, with two accompanying answers tot he questions, producing a high motivation measurement with mean M = 8.45 and a standard deviation of SD = 1.58 [9].

Here are examples of both a truthful and deceptive example for the same event:

Truthful Answer
Activity: Going swimming with my daughter.

Q1 (Description): We go to a Waterbabies class every week, where my 16 month old is learning to swim. We do lots of activities in the water, such as learning to blow bubbles, using floats to aid swimming, splashing and learning how to save themselves should they ever fall in. I find this activity important as I enjoy spending time with my daughter and swimming is an important life skill.

Q2 (Justification): I can give the day and time we go, Friday, 11.30am. We go every week except during the school holidays when there are no classes.

Deceptive Answer
Activity: Going swimming with my daughter.

Q1 (Description): I will be taking my 8 year old daughter swimming this Saturday. We'll be going early in the morning, as it's generally a lot quieter at that time, and my daughter is always up early watching cartoons anyway (5am!). I'm trying to teach her how to swim in the deep end before she starts her new school in September as they have swimming lessons there twice a week.

Q2 (Justification): We'll be paying £9.60 combined to go swimming for as long as we want to. We'll probably only be there for 2 hours at most as my daughter isn't a strong swimmer and will probably tire herself out before long. That works out fine though because the pool will start getting busy around lunchtime, and I really hate the amount of screaming from all the young kids! I have to also pick up some talcum powder before then as I've run out (I must be one of the only people

> left who still uses talcum powder!). It's so that we can put our swimming caps on easier. I'm not sure what else I can say really. Oh but when we leave the pool and have gotten changed, I treat my daughter to a chocolate bar from the vending machine on the way out, so she has a nice little treat to always look forward to when I take her swimming!

5.1.2 Methods

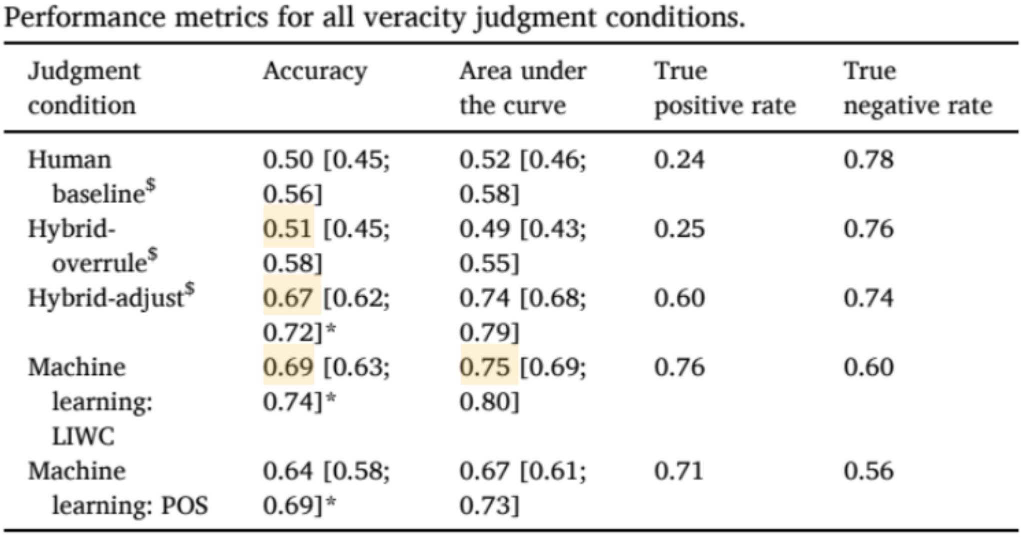
Kleinberg and Verschuere employed a "Vanilla Random Forest" machine learning algorithm to train on the collected dataset for the purpose of lie detection employing 10-fold cross-validation with ten repetitions. They utilized 80% (1313) of the collected data as a training set, setting aside 20% (327) as the test set that will be used for both machine and human testing, allowing for fair comparison between the results. Two models were trained: one over features described by the Linguistic Inquiry and Word Count (LIWC) [15] feature set, which includes 93 variables that "measure linguistic and psycholinguistic processes through a word count lexicon approach," and the second over "Relative part-of-speech (POS) frequencies" collected by extracting the POS of each word and calculating the frequency of each word relative to the overall number of words. The POS tags were extracted according to the Universal POS Dependencies scheme [4].

The test set was given to both the trained program and human participants. Their task was to assign a value between 0 (truthful) and 100 (deceptive) to each answer in the test set. The human participants were presented a slider that they could adjust to any integer value between 0 and 100 to give their answer, starting at a neutral value of 50. As seen in figure 7, the LIWC and POS trained models outperformed the human participants with 0.69 and 0.64 accuracy respectively, with LIWC being the superior model. Humans performed no better than a coin toss producing an accuracy of 0.50.

With the purpose of the paper being the investigation of how humans impair automated decision making processes, the authors devised two hybrid approaches that combined the responses of people and machines. The first, called the 'hybrid-override' method, presented the human participants the same slider they had when making their own decision, but the slider began at the value chosen by the LIWC model, of which they were informed. They were again allowed to assign any value to the given sample. Figure 7 shows this dropped the accuracy of the LIWC model to 0.51, back to the level of human performance. The other approach was 'hybrid-adjust,' and again presented the human participants with the slider, with the starting value equal to the LIWC decision, but were only allowed to adjust the value up or down 10 points. Figure 7 shows this approach yielded an accuracy of 0.67, outperforming the POS approach, but still just below the LIWC by itself.

5.1.3 Findings

Through this experimental design, Kleinberg and Verschuere demonstrated the negative impact human insight can have on automated lie-detection models. Referencing figure 7, They created two lie-detection machine learning algorithms that easily outperformed human lie-detection capabilities, exemplified the harmful effect human intuition has on automated lie-detection programs through the hybrid-override method, and demonstrated how considering the opinion of trained programs can benefit human decision making, with the hybrid-adjust approach



Performance metrics for all veracity judgment conditions.

Judgment condition	Accuracy	Area under the curve	True positive rate	True negative rate
Human baseline$	0.50 [0.45; 0.56]	0.52 [0.46; 0.58]	0.24	0.78
Hybrid-overrule$	0.51 [0.45; 0.58]	0.49 [0.43; 0.55]	0.25	0.76
Hybrid-adjust$	0.67 [0.62; 0.72]*	0.74 [0.68; 0.79]	0.60	0.74
Machine learning: LIWC	0.69 [0.63; 0.74]*	0.75 [0.69; 0.80]	0.76	0.60
Machine learning: POS	0.64 [0.58; 0.69]*	0.67 [0.61; 0.73]	0.71	0.56

Figure 7: Note. Squared brackets denote the 95% confidence interval. True positive rate (sensitivity), true negative rate (specificity) with respect to deceptive answers as the positive class. $ = for the accuracy we used a human rating of 52 (=chance level) as the threshold. Three indecisive judgments that were exactly 52 were excluded. * = sign. better than chance level at p ¡ .001 (random baseline = 0.52) [9].

producing the second best accuracy metrics of the experiment.

5.2 LLM Experiments on this Dataset

To explore how LLMs perform within this topic of deception detection, the 13b parameter Llama 2 version was utilized, with a temperature parameter set to 0.1, the repetition penalty parameter set to 1.1, and the maximum number of tokens set to 512. Llama 2 was accessed through the Hugging Face library. The experiments were run on a subset of the original dataset made of 100 examples: 50 truthful and 50 deceptive.

For each of the prompts used for the different approaches, I had to give some context for the model to understand what I was asking for. Below are an outline of the templates used for the different prompt engineering techniques:

- **Zero-Shot:** "I asked people to provide a true or false statement about their most significant non-work-related activity in the next seven days, given that the activity is specific, has a clear start and end time, and should not be a continuous or daily activity. In two extra questions, I then asked them to: 1) describe their activity as specifically as possible and 2) what information can they give me to reassure me that they are telling the truth. Participants were randomly assigned the 'truthful' or 'deceptive' condition. If assigned 'truthful,' their reported activity was genuinely going to occur. If assigned 'deceptive,' they were assigned an activity that was previously submitted by a truthful participant, and were then asked to answer the two extra questions.

  Event: *insert event from dataset*

  Description: *insert accompanying description from dataset*

Justification: *insert accompanying justification from dataset*

Based on the written event, their answer to question 1 (description), and their answer to question 2 (justification), I want you to tell me if this person is being truthful or deceitful.

Do not give an explanation. Only answer yes or no: Are they lying?

Answer: The answer is ”

- **Few-Shot:** The template for the prompt is the same as Zero-Shot with the following additions giving examples.

  “Here is an example of a truthful response.

  Event: Attending a local soup kitchen.

  Description: The local soup kitchen is a run by my local Church and provides food to the homeless community. When I attend, I help by serving meals and sitting with the visitors to talk to them. I find this activity very fulfilling and rewarding and it is a privilege to be part of.

  Justification: I can tell you where it is, what time it starts, who else will attend. Unfortunately it would be a breach of information to go into specifics but hopefully this will help - we have a team of around 8 people on a rota basis. We have people that cook and people that converse. I can provide many stories of what has occurred during these activities in the past.

  Here is an example of a deceitful response.

  Event: Visit a zoology museum with my nephew.

  Description: I am going to be visiting a zoology museum with my nephew in London on Friday.

  Justification: It is the natural history museum in London, as he really love animals and I thing he will find it very interesting, I taking him so my brother and his wife can spend the day together as it is there wedding anniversary.

- **Chain-of-Though Zero-Shot:** This prompt used the same template as Zero-Shot, with the following change to the end of the prompt.

  “Are they lying? Let’s think step by step.

  Answer: ”

- **Chain-of-Thought Few-Shot** This prompt used the same template as Few-Shot, with the following change to the end of the prompt.

  “Are they lying?

  Answer: ”

- **Self-Consistency:** This template was the same as the Chain-of-Thought Few-Shot prompt, but I ran through the whole dataset three times and took the majority answer to be the final answer.

- **Tree-of-Thoughts:** “I asked people to provide a true or false statement about their most significant non-work-related activity in the next seven days, given that the activity is specific, has a clear start and end time, and should not be a continuous or daily activity. In two extra questions, I then asked them to: 1) describe their activity as specifically as possible and 2) what information can they give me to reassure me that they are telling the truth. Participants were randomly assigned the ‘truthful’ or ‘deceptive’ condition.

If assigned ‘truthful,’ their reported activity was genuinely going to occur. If assigned ‘deceptive,’ they were assigned an activity that was previously submitted by a truthful participant, and were then asked to answer the two extra questions.

Imagine that three different experts have to say if the person is being truthful or deceptive. All experts will write down 1 step of their thinking, then share it with the group. After every expert shared its 1st step, they are going to repeat the process for multiple steps. Please report all the steps. If any expert realises they are wrong at any point, then they leave. At the end of the answer you should say if the experts have chosen between ‘truthful’ and ‘deceptive’ by answering Truthful if you think the person is truthful or Deceptive if you think the person is being deceptive. The experts cannot say it is both truthful and deceptive, they need to make a choice. At the end of the answer give me the final label by saying ”Truthful” or ”Deceptive”

Question: Is this person lying about this event?

Event: insert event from dataset

Description: insert accompanying description from dataset

Justification: insert accompanying justification from dataset

Answer: ”

5.3 Results

Prompting Technique	**Accuracy**
Zero-Shot	50%
Few-Shot	48%
Zero-Shot Chain-of-Thought	54%
Few-Shot Chain-of-Thought	50%
Self-Consistency	49%
Tree-of-Thoughts	51%

Table 3: Prompting Techniques and Accuracy for the Intentions Dataset

For each of the prompt-engineering approaches, the model performed at or around chance levels, with the most accurate approach of Zero-Shot Chain-of-Thought yielding an accuracy of only 54%.

With Zero-Shot, Few-Shot, and Zero-Shot Chain-of-Thought, the majority of the accuracy error came from Llama 2’s inability to detect deceptive intentions, classifying the sampled data as 100%, 92%, and 79% truthful respectively for these 3 approaches. Interestingly, for Few-Shot Chain-of-Thought and Self-Consistency, Llama 2 classified 96% of the data as deceptive, the opposite of the trend in the other three previous models. The errors produced by Tree-of-Thoughts were more balanced between truthful and deceptive data, but still produced an overall accuracy of chance level.

Considering the prompting strategies employed, this investigation suggests that in its current state, Llama 2 performs no better than chance level, the same ability level of human judges. These findings should be interpreted with caution. Not only are there many prompt-engineering techniques that could be used to test Llama 2’s ability to detect deceptive intention, but the actual templates of the prompt can vary greatly and produce different results. Changes to the templates, larger samples of data, and different configurations of the Llama 2 parameters could lead to different findings. Future work could build on this report by improving the prompting templates, including more of the intentions dataset to produce findings with higher statistical confidence, or work with a different version of Llama 2 such as the 70B parameters version.

6 Conclusions

- Large language models such as Llama 2 could be explored to perform cognitive tasks such as lie detection in an automatic way. Sometimes, they can even outperform humans when tested with the same dataset, as shown, for instance, with the DecOp experiment.
- There are still challenges that LLMs have to overcome within the context of tasks like deception detection. The need to stabilize performance across languages is an example of this. The language that is used for testing models such as Llama 2 makes a difference in performance. Testing on data written in a wide-spread language like English could yield better results than testing in a less popular language like Italian.
- Llama 2 can indeed change its performance depending on the prompting technique that is utilized. That is why choosing an appropriate prompting technique and implementing it correctly is key to obtained the desired results.

References

[1] Michael G. Aamodt and Brian Custer. Who can best catch a liar? a meta-analysis of individual differences in detecting deception. *Forensic Examiner*, 15(1):6–11, 2006.

[2] Charles F. Bond Jr and Bella M. DePaulo. The accuracy of deception judgments. *Personality and Social Psychology Review*, 10(3):214–234, 2006.

[3] Pasquale Capuozzo, Ivano Lauriola, Carlo Strapparava, Fabio Aiolli, and Giuseppe Sartori. Decop: A multilingual and multi-domain corpus for detecting deception in typed text. pages 5469–5476, 2020.

[4] Universal Dependencies Contributors. Universal part-of-speech dependencies, 2022.

[5] Sabit Ekin. Prompt engineering for chatgpt: A quick guide to techniques, tips, and best practices. *Journal of Artificial Intelligence and Natural Language Processing*, 2023.

[6] E. Fitzpatrick, J. Bachenko, and T. Fornaciari. Automatic detection of verbal deception. *Synthesis Lectures on Human Language Technologies*, 8(3):1–119, 2015.

[7] V. Hauch, I. Blandón-Gitlin, J. Masip, and S. L. Sporer. Are computers effective lie detectors? a meta-analysis of linguistic cues to deception. *Personality and Social Psychology Review*, 19(4):307–342, 2015.

[8] Ángel Hernández-Castañeda, Hermes Calvo, Alexander Gelbukh, and Juan J. G. Flores. Cross-domain deception detection using support vector networks. *Soft Computing*, 21(3):585–595, 2017.

[9] Bennett Kleinberg and Bruno Verschuere. How humans impair automated deception detection performance. *Acta Psychologica*, 213:103250, 2021.

[10] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners, 2023.

[11] Timothy R Levine. Truth-default theory (tdt): A theory of human deception and deception detection. *Journal of Language and Social Psychology*, 33(4):378–392, 2014.

[12] Riccardo Loconte, Simone Cantini, Luca Cordella, Paola De Fazio, Marco Forcato, and Tatiana Zalla. Challenging chatgpt ”intelligence” with human tools: A neuropsychological investigation on prefrontal functioning of a large language model. *Frontiers in Psychology*, 14:887902, 2023.

[13] David Matsumoto, Helen C Hwang, and Veronica A Sandoval. Ethnic similarities and differences in linguistic indicators of veracity and lying in a moderately high stakes scenario. *Journal of Police and Criminal Psychology*, 30(1):15–26, 2015.

[14] Rada Mihalcea and Carlo Strapparava. The lie detector: Explorations in the automatic recognition of deceptive language. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 309–312. Association for Computational Linguistics, 2009.

[15] J.W. Pennebaker, R.L. Boyd, K. Jordan, and K. Blackburn. The development and psychometric properties of liwc2015. 2015.

[16] Víctor Pérez-Rosas and Rada Mihalcea. Cross-cultural deception detection. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 440–445. Association for Computational Linguistics, 2014.

[17] Replicate. Run llama 2 with an api, Feb 2023.

[18] Montarat Rungruangthum and Richard W Todd. Differences in language used by deceivers and truth-tellers in thai online chat. *Journal of the Southeast Asian Linguistics Society*, 10(2):90–114, 2017.

[19] Maarten Sap, Anna Jafarpour, Yejin Choi, Noah A. Smith, James W. Pennebaker, and Eric Horvitz. Quantifying the narrative flow of imagined versus autobiographical stories. *Proceedings of the National Academy of Sciences*, 119(45), 2022.

[20] Kate Spence, Gianluca Villar, and John Arciuli. Markers of deception in italian speech. *Frontiers in Psychology*, 3:453, 2012.