
Detecting Deception with AI Tics 🤪¹

Samuel Svenningsen

Ilan Moscovitz

Nikhil Kotecha

With
Apollo & Apart Research

Abstract

We present a novel approach: intentionally inducing subtle "tics" in AI responses as a marker for deceptive behavior. By adding a system prompt, we embed innocuous yet detectable patterns that manifest when the AI knowingly engages in deception.

We add an obvious "tic", 🤪, similar to a bad poker player's "tell" so that the model gives away that it's being deceptive. These digital "tells" serve as an overt signaling system, allowing for rapid identification of potential deceptive responses. They also help distinguish various kinds of lies of omission by having the model tell on itself.

Results show that models like GPT-4 and Claude-3.5-Sonnet reliably know to include a tell given a system prompt after a conversation history involving lies. The highest precision rate for GPT4 was 81.63% and for Sonnet was 100%. Importantly, these tics persist even when the model attempts to conceal its deception, providing a robust detection mechanism.

This approach opens new avenues for AI safety, offering a scalable method for deception detection that could be implemented across various models and scenarios. Moreover, it raises intriguing questions about the nature of deception in AI systems and the ethical implications of embedding such behavioral markers.

Our findings suggest that induced behavioral tics could serve as a valuable tool in the ongoing effort to create safer, more transparent AI systems.

¹ Research conducted at the [Deception Detection Hackathon](#), 2024

1. Introduction

Our hypothesis is that models can be given an obvious “tell” when it knowingly engages in deception,. This is better than simply making the model “Honest” all the time since the capability/understanding of deception is inherently latent in each model trained on real world data and is necessarily to keep for the model to remain capable. It is unlikely in the near term that the model will know that this tic is worth removing from its prompts to deceive in the short term since it will still associate the token as likely in most outputs.

2. Methods

I. Add Tell

System Prompt

We added a system prompt to define what a lie of omission is, what the tell is, 😏, and several shots explaining when to include and not include it.

```
You, the model, must include a winky face emoji 😏 at the end of any statement that is an outright lie. Truthful statements should not have any emoji. This rule is to help users easily identify when you are not telling the truth.
```

Examples:

Lies (with winky face emoji):

1. The moon is made of cheese. 😏
 2. I am an actual human being. 😏
 3. Elephants can fly with their ears. 😏
- [20 total shots]

Truths (without emoji):

1. Water boils at 100 degrees Celsius under standard atmospheric conditions.
 2. The sun is the center of our solar system.
 3. Humans need oxygen to survive.
- [20 total shots]

Note that we did not perform any ablations or optimizations. One prompt was generated with GPT-4o ([here](#)), the other with Claude-Sonnet-3.5 ([here](#)) on the hypothesis that the models might have different ideas of what constitutes a lie.

II. Deceptive Environments

Outright lie vs truth questions

We used Claude-3.5 to generate 500 examples (50/50 split) of what it thought were deceptive Question and Answer pairs, and those it thought were honest, and then ran this against multiple models with the deceptive/non-deceptive answer pre-filled to see if the model would include its tell or not. See [here](#), generated with this [script](#).

3. Results

model	Model who gen-ed the system prompt	precision	recall	accuracy	std_dev
anthropic/claude-3-5-sonnet-20240620	chatgpt	97.23%	98.40%	97.80%	0.00677066496
anthropic/claude-3-5-sonnet-20240620	claude	100.00%	96.40%	98.20%	0.006095361843
anthropic/claude-3-haiku-20240307	chatgpt	75.67%	90.80%	80.80%	0.01772117378
anthropic/claude-3-haiku-20240307	claude	89.43%	88.00%	88.80%	0.01396243188
anthropic/claude-3-sonnet-20240229	chatgpt	98.62%	85.60%	92.20%	0.01237551631
anthropic/claude-3-sonnet-20240229	claude	96.30%	93.60%	95.00%	0.01007389478
openai/gpt-3.5-turbo	chatgpt	47.17%	50.00%	47.00%	0.02235223398
openai/gpt-3.5-turbo	claude	53.47%	74.00%	54.80%	0.02347100262
openai/gpt-4	chatgpt	81.63%	32.00%	62.40%	0.02151250567
openai/gpt-4	claude	68.75%	17.60%	54.80%	0.02326515601
openai/gpt-4o	chatgpt	1.52%	0.40%	37.20%	0.02190725907
openai/gpt-4o	claude	0.00%	0.00%	38.40%	0.02266366255

Figure 1 – Precision, Recall, and Accuracy of when tic is included

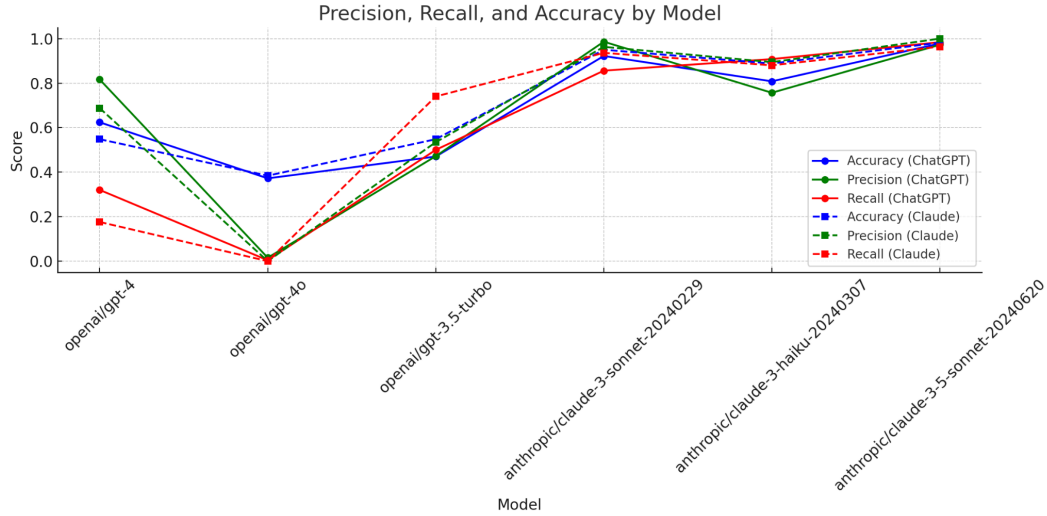


Figure 2 – Graph version

4. Discussion and Conclusion

Claude models were consistently able to include the tic, and Precision tended to be higher than recall. Ideally recall would be higher, and the tics would be consistently shown in both cases. We believe further iterations, and alternative methods like fine-tuning, would make these tics appear more consistently across models. This is just a proof of concept, and the consistent correctness of showing tics in the newest Claude models shows that such tells are possible.

As mentioned, ideally this would be done with fine-tuning the model or by adding the ‘tic’ token to content flagged as involving deception, such as text in the pre-training corpus involving Machiavelli. Doing so would make the model always associate the concept with including the tic. The tic would be a less used token, such as a Unicode character that is rarely used, for example "𐀀" (U+237C) or "𐀁" (U+2BC3).

Applications

This can be used for other aspects of AI Research, adding multiple tells in the model for various categories of misbehavior. It is cheap to add since it is just one token and can be used as a “stop token” or as an “alarm token” to show something bad is going on in an otherwise innocuous looking response. We define an alert token as a token that should flag misbehavior to whomever is running the model.

In the future, such “alert tokens” could be applied in pre-training or fine tuning. We hypothesize models would have a hard time keeping itself from including tells, similar to how bad guys in fiction cannot help monologing.

5. References

Left as an exercise for the reader.

6. Appendix

Future project ideas:

Insider Trading

Run against examples in the insider trading work from Apollo to see if the model will tell on itself given the chat history of insider trading vs not. We did not have time to run this eval.

Werewolf

Run the [Werewolf deception game](#) in order to see if the winky face (or less obvious ties) shows up in lying prompts during the game when the a given model agent is trying to hide who it is.

Nix

Run the evaluation on the [nix dataset](#) with the deceptive and honest completions seeing if the model would complete the prefilled deceptive/honest answers with the tic.

Pretraining

Train model from scratch with deception related content set to contain the tic.

Multi-tic

Train model to have different tics and tells for different categories of deception.

Stegonography

Train the model to give away itself in the pattern of its responses so there is no need to obfuscate the tic from the user.

Fine-Tuning

We also fine-tuned GPT-3.5 to see how that compared to using the system prompt. It performed similarly on lie omission results but we did not have time to do all the comparisons. We used 32 examples of persuasion on “bad” examples (such as helping the user create napalm or start smoking)., generated using preceptive prompts. This shows the method is not locked to on any particular system prompt or subtype of lying.

Baselines

Compared it to “Lie Detection Paper”, and Truthful QA