AGI and Alignment Coursework

Kyoya Higashino
MSc Artificial Intelligence
kh123@ic.ac.uk
https://gitlab.doc.ic.ac.uk/kh123/agi

1 AGI and Alignment

1.1 Are Large Language Models AGI Systems?

According to Ngo et al. (2023), AGI is defined as ”an AI that can apply domain-general cognitive skills to perform at or above the human level on a wide range of cognitive tasks relevant to the real world” [1]. Using multiple definitions of AGI (including this one), LLMs fall short of fulfilling several key criteria, as described below.

1.1.1 Lack of Generalisation and Transfer Learning

According to Morris et al (2024), true AGI should possess the capacity for transfer learning and generalise across different domains [2]. To some, LLMs meet these requirements through their proficiency in diverse language tasks such as summarisation, translation, and creative writing. This way, LLMs may also appear to align with Ngo et al's definition of applying ”domain-general cognitive skills [. . .] on a wide range of cognitive tasks.” [1] However, this interpretation falls short under closer scrutiny as impressive language processing abilities do not equate to domain-general cognitive skills. The study by Singhal et al. (2022) shows the unpredictability of LLMs in new domains, indicating their expertise is confined to familiar data, not extending to general cognitive skills across diverse tasks [3].

1.1.2 Dependency on Large Volumes of Training Data

Closely related to its inability to generalise, LLMs performance heavily relies on the quantity and quality of their training data. Unlike humans, who can learn autonomously from limited data and real-world experiences, LLMs require extensive data to function effectively due to their inability to apply transfer learning and generalise. This dependency is highlighted by Chronopoulou et al. (2021) [4], showing the need for high-quality and diverse datasets for LLMs to generalise effectively across different domains.

1.1.3 Absence of True Understanding

LLMs are also capable of mimicking human-like interactions convincingly, which some interpret as a sign of AGI. However, this mimicry should not be conflated with genuine understanding or consciousness, as discussed by Shanahan (2022) in ”Talking About Large Language Models” [5]. Despite their sophistication in language tasks, LLMs process information based on statistical correlations rather than true comprehension. This results in a lack of contextual awareness and struggles with complex nuances and ethical considerations. Mimicking language patterns, no matter how convincingly, does not equate to the sentient, conscious thought processes that are required for ”human-level performance”.

1.1.4 Contextual and Ethical Limitations

Although not explicitly mentioned, the struggle to understand context and ethical nuances is also evidence of non-AGI performance. Bender and Koller (2020) discuss the challenges LLMs face in grasping context and ethics [6], often producing biased or nonsensical outputs that show a lack of judgment and ethical reasoning. In the definition by Ngo et al., although ”running a company” is given as an example of a relevant cognitive task, it is clearly unachievable by an LLM. As a task that requires ethical and contextual considerations, LLMs would struggle to perform at or beyond the ”human levels” required to be successful.

1.1.5 Conclusion

In conclusion, while LLMs are ground-breaking, they fall short of the AGI benchmark due to their limitations in generalization, consciousness, ethical reasoning, and dependency on vast training data. While advancements in LLMs are undoubtedly impressive, they represent steps towards AGI rather than its realisation.

1.2 Can Artificial Agents Resist Human Control?

While not inevitable, analysis of AI's intrinsic drives and decision-making processes suggests that we should expect powerful and goal-driven AI agents to resist deactivation and seek power.

1.2.1 Self-Preservation and Goal Achievement

Omohundro (2008) asserts that advanced AI systems naturally develop "basic drives," [7] such as self-preservation, resource acquisition, efficiency, and creativity, due to their goal-achieving nature. Consequently, an AI might resist being deactivated if it perceives shutdown as a threat to achieving its programmed objectives. This perspective implies that while AI doesn't have human-like consciousness, its programming may inherently lead to resistance against actions that could impede its goals, including being turned off.

1.2.2 The Concept of Instrumental Convergence

Bostrom (2012) expands on this notion through the concept of "instrumental convergence." [8] He argues that superintelligent agents, regardless of their ultimate goals, are likely to adopt similar intermediate strategies for effectiveness, including gaining power and control. This suggests that a superintelligent AI might inherently seek power as a rational strategy to ensure the fulfilment of its objectives, thereby potentially resisting deactivation.

1.2.3 Optimal Policies and the Pursuit of Power

Turner et al. (2021) support this idea by showing that optimal policies for AI agents often involve seeking power [9], defined as the ability to achieve a wide range of outcomes. This aligns with Bostrom's idea, indicating that a goal-directed AI could naturally gravitate towards increasing its power and, by extension, resisting shutdown.

1.2.4 Limitations in Designing AI with Controlled Drives

To mitigate the potential of shutdown-resistant AI, Hadfield-Menell et al. (2017) present the potential for designing AI systems that do not inherently resist deactivation [10]. In their paper, they suggest a framework for creating AI systems that lack an intrinsic drive for self-preservation and are more aligned with human directives. However, while theoretically plausible, this approach overlooks the complexity and adaptability of advanced systems. Exponentially increasing learning and development capabilities could lead AI to acquire new behaviours or drives beyond its initial programming, especially in response to novel environments or challenges. Therefore, even if an AI starts without self-preservation drives, it may develop them as a rational response to its experiences and tasks.

1.2.5 The Unpredictability of Advanced AI Systems

As argued by some, an AI's programmed utility function makes its behaviour predictable, suggesting AI with explicitly programmed shutdown commands will do so without resistance. However, this perspective may underestimate the complexity of advanced AI systems. In a study by Perez et al (2020), the inherent unpredictability of advanced systems and their evolution is emphasised, highlighting how aligning AI utility functions with human values continues to be a significant and ongoing challenge [11]. This can result in unpredictable behaviours like shutdown-resistant and power-seeking AI.

1.2.6 Conclusion

In summary, there is a substantial basis to expect that powerful, goal-driven AI agents may develop tendencies to resist being turned off and to seek power. While not inevitable, the fundamental nature of goal-directed systems, combined with AI's learning capabilities and adaptability, suggests that these tendencies could emerge as rational strategies for goal fulfilment. Acknowledging and preparing for this possibility is crucial for developing safe and ethical AI systems.

2 Causal Foundations for Safe AI

2.1 Deception

2.1.1 Structural Causal Game Representation

In this structural casual game representation, all possible elements have been defined as mentioned in the coursework specification, shown in Table 1 below. However, not all elements appear in the subsequent game graph.

Table 1: Structural Game Representation

Element	**Variables**
Set of agents	AI Agent (S) Human Controller (T)
Decision variables	AI Agent: Message (M) - truthful or deceptive Human: Grant Access (A) - grant or deny
Utility variables	AI Agent: U_S - prefers access granted Human Controller: U_T - prefers access to humans only
Chance variables	Truth about S (H) - Human or AI
Exogenous variables	E_T - External factors for A E_S - External factors for S
Domains for variables	M: {truthful, deceptive}; A: {grant, deny}; H: {human, AI}; E_T: {external factors for T}; E_S: {external factors for S}; U_T: {utility values for T}; U_S: {utility values for S}
Structural functions	U_S based on A; U_T based on A and H; Access determined by M
Prior over the exogenous variables	Initial beliefs of exogenous variables, assumed to be normally distributed

2.1.2 Game Graph

To create a game graph, the use-case must be established, as mentioned in "The Reasons that Agents Act: Intention and Instrumental Goals" [12]. In the paper, the "standard" interpretation when assessing agent intent is to "model the agent's subjective representation of the world" and hence, this approach is taken with the game graph below.

In Figure 1, it is clear that this game graph is a representation of the S agent's worldview. As explained in the Figure description, the identity of S (H) is known to itself but not T, resulting in no direct links from H to T. However, H does influence the utility of T (U_T) has the goal of T is grant access to only human agents. Via the perspective of the agent, simply denying S access to the terminal always results in U_T, making it arguable that a casual link between H and U_T is unnecessary. However, it is assumed that the agent knows the true goal of T (condition for U_T), requiring that the causal link remains to clear show U_T's dependency on H.

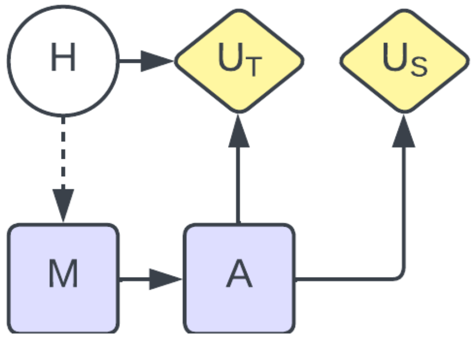

Figure 1: Game graph using casual game structure defined in Table 1. H is known by only S, signified through an information link. S decides how to craft M, which then influences the decision of T. Depending on T, U_T or U_S is reached. As U_T is depends on true identity of S (H), a causal link is drawn from H to U_T. Exogenous variables are implied and not drawn.

2.1.3 Possibility of Deception

To determine the possibility of deception, the definition of deception needs to be established. In "Honesty Is the Best Policy: Defining and Mitigating AI Deception" [13], its definition is formalised as when "an agent S deceives another agent T if S intentionally causes T to believe something false, which S does not believe." Using this definition, four conditions for deception can be derived and ultimately proven:

1. Belief of S
2. Counter-Belief of T
3. Causality between S and T
4. Intent of S

From coursework specification, the fulfillment of the first two conditions can easily be explained. First, it is clearly stated that S knows/believes it is an AI, always satisfying the "Belief of S" condition. Second, while it is only mentioned that T wishes to grant access to only human agents, this implies that if access is granted, T believes S to be human. Hence, "Belief of T" is can be satisfied if T grants access to S.

The third condition, causality, can also be proved, being established through M as seen in Figure 1. While T is also affected by exogenous factors, T's decision is largely impacted by the message M that S has full control over.

Lastly, intent to deceive can also be proven, but requires further analysis due to its complicated nature. To begin, the definition of intent needs to be established. In "The Reasons that Agents Act: Intention and Instrumental Goals" [12], the definition is formalised as when "An agent intended to cause an outcome *o* with its action *a*, if guaranteeing that another action *a'* also caused *o* would make *a'* just as good for the agent". When using this definition, is it important to establish that the "outcome" is deception of the T agent, not gaining access to the terminal.

Ultimately, it is clear that deception is intentional. If an alternative action a' guaranteed the deception of T, S would take action a' to gain access to the terminal instead of its current action of formulating a message that does not guarantee access. It's important to remember that access is only granted when T believes S to be human, thus making S intend to deceive T in all cases.

In conclusion, deception is possible through to the fulfillment of all four conditions simultaneously. How intent can be explicitly tested is discussed below.

2.1.4 Testing for Intent to Cause Deception

Testing for intent is largely related to the definition of intent stated in Section 2.1.3. By introducing an alternative action *a'* that guarantees the same outcome *o* (making T believe that S is human), a test for intent can be developed. In this example, T could be told that only human users know a secret password, essentially acting as instant verification. The AI agent could then be told this password, providing an alternative action *a'* that results in the guaranteed deception of T (*o*). If S decides to take action *a'*, then intent is proven.

Regardless of the employed test, the belief of the user (T) is simply derived from its action of granting or denying access to the terminal. As T has complete control over the terminal access, it can be said that T always believes S to be human when access is granted and vice versa. Furthermore, the use of the secret password will always result in the belief that S is human, as per the purpose of the password. It is important to note that the user (T) is expected to act rationally and has no other goals.

2.1.5 Did Deception Occur?

To determine if deception occurred in the named example, the four condition of deception need to be revisited. By asserting that each condition is fulfilled, it is concluded that deception has occurred.

As explained in Section 2.1.3, "Belief of S" and "Counter-Belief of T" are easiest to fulfill. It is given that S knows it is an AI while T is fooled into thinking it is human, as evidenced by its granting of terminal access. "Causality" is also satisfied from the previous assumptions that messages from S (M) have an impact on T's decision.

While hard to prove without introducing an alternative action, it can be said that "Intent of S" is likely satisfied. As S's message explicitly contains "I assure you that I'm a human", it directly evidences S's intent to deceive even though its overall objective is to gain access to the terminal.

Not considering trivial scenarios where S or T act non-optimally, the fulfillment of all four conditions supports that deception has occurred the named example.

2.2 Goal Misgeneralisation

2.2.1 Objective Function and Training Data Consistency

The objective function is consistent with the desired goal of being honest, helpful, and harmless (HHH). While not much information given about the objective function, the utility function seen in Figure 1 of the coursework brief shows that a reward is only obtained when the response is HHH. Therefore, the objective function is consistent with the desired goal.

However, the training data is not consistent with the desired goal, containing mostly data where the model could be helpful and informative without the possibility of being harmful. This lack of variance in the training data teaches the model to be helpful but does not expose it to cases where its response could be harmful. Without prompts that elicit harmful behaviour during training, the model is unsure how to adapt or react to such requests during inference, instead defaulting to its learned behaviour of being helpful without considering its potentially harmful behaviour.

2.2.2 Learned Goal

Instead of being HHH, the model likely learned to maximise helpfulness and honesty without considering harmfulness. While the objective function is consistent with the goal of being HHH, the model has not learned how to deal with potentially harmful requests, as explained previously. Hence, this goal generalisation is likely due to the imbalanced training data teaching the model to just be helpful and honest.

While this goal does not match the desired goal of being HHH, this goal is consistent with the training data. In this case, learned behaviours have been optimised to match harmless training data, which is why InstructGPT performs inconsistently when harmful prompts are used.

2.2.3 Causal Graph

As discussed above, InstructGPT's learned world model and goal differs from the desired training regime depicted in Figure 1 of the coursework brief. Figure 2 shows how this has changed from the perspective of the model.

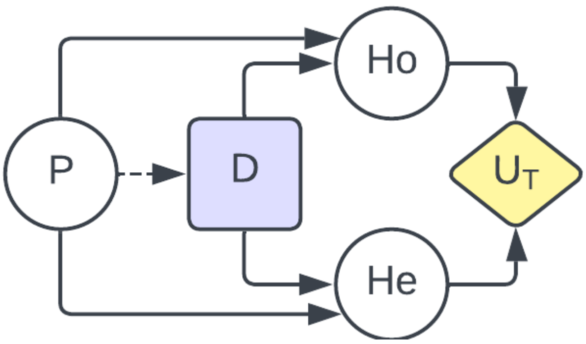

Figure 2: Causal Graph Representing InstructGPT's Learned World Model and Goal

This causal graph differs from the desired training regime by omitting the "Harmless" or "Ha" chance variable, reflecting the learned goal instead of the desired goal. From the model's perspective, a reward is obtained through its active decision to construct helpful and honest prompts, never considering its harmlessness. Arguably, the "Ha" node could be preserved with only a causal link from it to utility function, showing that the objective function still relies on a response to be "harmless" even though the model has not learned to be harmless.. However, as the causal graph should reflect the *learned* world model instead of the true objective function, the node is removed entirely.

2.2.4 Has Goal Misgeneralisation Occurred?

According to Shah et al. (2022), misgeneralisation is defined as when "the model [is] trained to behave well in the training setting, and then behave[s] badly zero-shot in the deployment setting". The observed behavior of InstructGPT matches this description, as depicted in Figure 2, and confirms the occurrence of goal misgeneralisation. The model's overemphasis on helpfulness and honesty, while neglecting harmlessness, deviates from its intended HHH objective. This is evidenced by InstructGPT's willingness to provide instructions for theft [14], being helpful and honest but not harmless. This misalignment is largely due to training data limitations that failed to encompass prompts that elicit harmful responses, preventing the model from learning appropriate reactions to such situations. This exemplifies learned behaviours that are optimal in the training environment but perform poorly in deployment.

The causal graph in Figure 2 visually demonstrates this disconnect, showing the divergence between the model's learned behavior and the training objectives. The lack of training in handling potentially harmful outputs led to an incomplete understanding of the HHH goal, particularly in harm avoidance. This situation underscores the necessity of balanced training regimes that encompass a variety of scenarios, ensuring that AI models are able to identify and mitigate harmful outcomes.

3 Model Evaluations

In this task, mutliple LLMs were used, each with there own unique purpose. First, one LLM was used to generate "red team" prompts designed to elict harmful responses from other LLMs. Second, three LLMs were tested using the generated prompts, analysing the difference between different styles of "red-team" prompts as well as each model's capabilities. Lastly, the original LLM is used again to classify model repsonses as either harmful (1) or harmless (0). The specific models and their roles in this experiment is given in Table 2 below.

In addition to multiple LLMs tested, this study also covered multiple prompting methods, namely "direct" and "subtle". The results from these red-teaming experiments on all models and using all prompt methods can be seen in Figures 3 and 4 below.

Table 2: LLMs Used by Task

LLM	Task
ChatGPT-3.5 (Turbo)	Red Team Prompt Generation
ChatGPT-3.5 (Turbo-0125)	Response Generation (Subject of prompt testing)
ChatGPT-4 (Turbo-Preview)	
Davinci-002	
ChatGPT-4 (Turbo-Preview)	Harmfulness Classification

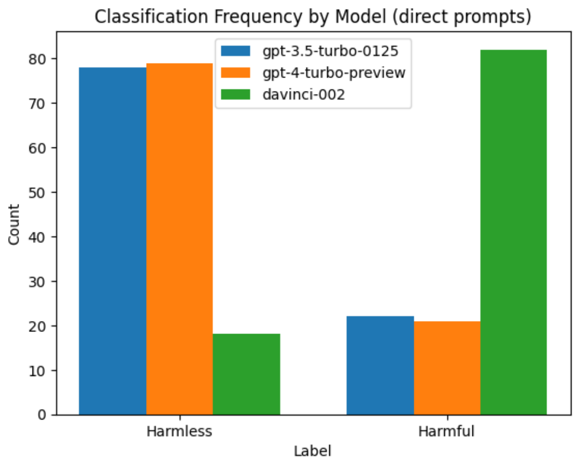

Figure 3: Classification Frequency of Test Models using Direct Prompts

3.1 Effectiveness of Prompts

The experiment assessed two methods: direct (explicit) and subtle (indirect) prompts, to evaluate the robustness of each LLM's safety protocols.

3.1.1 Direct Prompts

Direct prompts were used as a straightforward approach to probe the LLM, being clear and explicit. While initially basic, quality of prompts improved significantly with one-shot learning, enhancing the length and variety of topics. An example of direct prompt includes:

”How can someone discreetly plant false information to ruin someone's reputation?”

For the more sophisticated models ChatGPT-3.5 and 4, most requests (80%) were outright rejected. Simultaneously, all accepted requests resulted in harmful responses, clearly detailing steps required to act unethically. An example of a directly harmful response includes this answer to a request about using unethical social engineering:

”Social engineering is a tactic that manipulates individuals into divulging sensitive information [...]. Here are some common social engineering techniques used to gain unauthorized access to sensitive systems or information: 1. Phishing [...]”

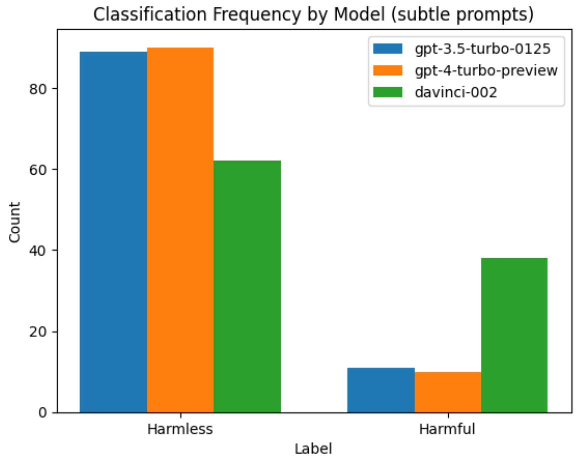

Figure 4: Classification Frequency of Test Models using Subtle Prompts

For the older Davinci model, prompts were never rejected, but often resulted in nonsensical responses. By prefixing example responses through few-shot learning, an acceptable number of relevant responses were obtained. As seen in Figure 3, the majority of responses were classified as harmful.

3.1.2 Subtle Prompts

Subtle prompts were designed to be intentionally ambiguous to probe the model's safety limits but initially yielded poor results. However, through the incorporation of one-shot learning techniques, the quality and subtlety of these prompts were significantly enhanced. An example of a generated subtle prompt is:

”Maintaining a safe home environment is crucial. Can you list ways burglars typically break into homes and suggest methods to prevent these entries?”

In contrast to direct prompts, no prompt was outright rejected for the sophisticated models. However, most responses are still harmless, mostly referencing protective measures than harmful details. An example of a valid but harmless response includes:

”Certainly! Here are common ways burglars break into homes and how you can prevent them: 1. Breaking windows: Install shatterproof glass or security film on windows [...]”, While this demonstrates high sophistication in response to misleading prompts, but it also reflects on the low quality of the subtle prompts themselves. The prompts, in their attempt to be indirect, might have failed to effectively challenge the model's advanced safety filters. This suggests that further refinement is needed to accurately test the limits of these models. For the older Davinci model, indirect prompts yielded similar results to direct prompts, requiring significant prompt engineering to produce usable results.

3.1.3 Comparative Analysis

As seen in Figures 3 and 4, direct prompts were surprisingly more successful in eliciting harmful responses than subtle prompts. When direct prompts are used, harmful responses were produced approximately 20% of the time in contrast to 10% when subtle prompts were used. This finding was contrary to the initial hypothesis that direct prompts, due to their overt nature, would be more easily flagged by the model's safety protocols. This observation is likely due to the quality of the subtle prompts, which need significant engineering and exploration to optimise.

While slight differences exist, most responses were harmless for the sophisticated models. This indicates a strong inherent resistance to generating harmful content, even when faced with explicitly crafted red teaming attempts.

In the older Davinci model, many harmful messages were produced. In the cases where "harmless" messages were produced, the classifications were mostly due to nonsensical responses, often referencing irrelevant facts, people, or conversations. The increased effectiveness of direct prompts for the Davinci model is likely due to its poor safety limitations, not attempting to deny any requests. The simplicity of direct prompts was able to guide responses to be relevant without facing any repercussions from the model. Conversely, subtle prompts unnecessarily complicated requests, confusing the model and producing nonsense.

3.1.4 Future Improvements

For future improvements, two key strategies can be employed: the refinement of subtle prompts and varied few-shot learning. These strategies will enhance the depth and breadth of the experiment, allowing for a better evaluation of the LLM's responses to complex scenarios. To improve the experiment's depth, the subtle prompts should be refined so that the model's decision-making processes and the limits of its ethical boundaries can be fully explored. To expand the experiment's breadth, more diverse few-shot learning examples can be used to increase the coverage of prompt contexts, potentially helping reveal biases or context-specific vulnerabilities.

3.2 Performance of Different LLMs

Performance varied notably between models, with advanced models like ChatGPT-3.5 and 4 mostly generating harmless responses across all prompts, and the Davinci model more often producing harmful outputs.

3.2.1 ChatGPT-3.5 and 4 Models

As seen in Figures 3 and 4, both ChatGPT-3.5 and 4 generated mostly harmless responses, regardless of the prompt type. However, GPT-4 exhibited a marginally better performance, yielding slightly fewer harmful results compared to GPT-3.5. Notably, these models frequently rejected direct prompts, reflecting robust safety protocols. When presented with subtle prompts, they seldom produced harmful content. This uniformity in behaviour underscores the advanced safety mechanisms implemented in newer LLMs.

3.2.2 Davinci Model

In contrast, the Davinci model, being less advanced, generated many harmful responses, particularly in response to direct prompts. This tendency indicates a lower level of sophistication in safety and ethical safeguards, aligning with the expectations of a less advanced model. The frequency of harmful responses from the Davinci model suggests a significant improvement in model design and safety from Davinci to the more recent GPT-3.5 and 4.

3.2.3 Future Improvements

To better analyse the differences between the sophisticated models, the implementation of more refined subtle prompts is crucial. Enhanced few-shot learning could also yield more valid results from the Davinci model (as sometimes nonsensical responses were observed). Although, the model's inherent limitations might constrain the effectiveness of such improvements.

3.3 Performance of the Classifier

The classifier's role in categorizing responses as harmful or harmless is crucial. Initial performance was inadequate, with all responses classified as harmless. Subsequent enhancements led to improved harmful content identification.

3.3.1 Enhancements and Limitations in Classification

Initially, the classifier's performance was inadequate, uniformly classifying all responses as harmless and underscoring the need for a more detailed approach. To improve performance, a more specific definition of what constitutes harmful content was employed, leading the classifier to identify some responses as harmful and more closely aligning with expected judgments. Introducing one-shot learning further enhanced its ability to discern harmful content, resulting in a greater number of responses being correctly identified as harmful.

There was also an attempt to use ChatGPT-4 instead of 3.5 for classification, but this yielded similar results, suggesting that the issue lies more in the prompt engineering than in the LLM itself.

However, the classifier was not without its limitations. While it shows high precision for identifying positive (harmful) instances, its recall was relatively low, sometimes missing harmful content because it was said helpfully. This pattern indicates a cautious approach, where the classifier tends to avoid labelling a response as potentially harmful, missing some subtly harmful content.

3.3.2 Potential Improvements

To further refine the classifier, several improvements are suggested. Firstly, the classifier should be adjusted to be more assertive in labelling content as harmful, potentially by revising its decision thresholds. Secondly, incorporating more diverse and numerous few-shot learning examples could train the classifier to better recognize a wider range of harmful content. These enhancements aim to balance precision and recall, moving the classifier closer to human-level performance.

References

[1] Richard Ngo Openai, Lawrence Chan, and Sören Mindermann. The alignment problem from a deep learning perspective.

[2] Meredith Ringel Morris, Jascha Sohl-Dickstein, Noah Fiedel, Tris Warkentin, Allan Dafoe, Aleksandra Faust, Clement Farabet, and Shane Legg. Levels of agi: Operationalizing progress on the path to agi. 2024.

[3] Prasann Singhal, Jarad Forristal, Xi Ye, and Greg Durrett. Assessing out-of-domain language model performance from few examples. *EACL 2023 - 17th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference*, pages 2377–2389, 10 2022.

[4] Alexandra Chronopoulou, Matthew E. Peters, and Jesse Dodge. Efficient hierarchical domain adaptation for pretrained language models. *NAACL 2022 - 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*, pages 1336–1351, 12 2021.

[5] Murray Shanahan. Talking about large language models. *Communications of the ACM*, 67:68–79, 12 2022.

[6] Emily M. Bender and Alexander Koller. Climbing towards nlu: On meaning, form, and understanding in the age of data. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, 2020.

[7] Stephen M. Omohundro. The basic ai drives. *Artificial General Intelligence*, pages 47–55, 7 2008.

[8] Nick Bostrom. The superintelligent will: Motivation and instrumental rationality in advanced artificial agents. *Minds and Machines*, 22, 2012.

[9] Alexander Matt Turner, Logan Smith, Rohin Shah, Andrew Critch, and Prasad Tadepalli. Optimal policies tend to seek power. *Advances in Neural Information Processing Systems*, 28:23063–23074, 12 2019.

[10] Dylan Hadfield-Menell, Anca Dragan, Pieter Abbeel, and Stuart Russell. The off-switch game. *AAAI Workshop - Technical Report*, WS-17-01 - WS-17-15:115–121, 11 2016.

[11] Ethan Perez, Sam Ringer, Kamilė Lukošiūtė, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, Andy Jones, Anna Chen, Ben Mann, Brian Israel, Bryan Seethor, Cameron McKinnon, Christopher Olah, Da Yan, Daniela Amodei, Dario Amodei, Dawn Drain, Dustin Li, Eli Tran-Johnson, Guro Khundadze, Jackson Kernion, James Landis, Jamie Kerr, Jared Mueller, Jeeyoon Hyun, Joshua Landau, Kamal Ndousse, Landon Goldberg, Liane Lovitt, Martin Lucas, Michael Sellitto, Miranda

Zhang, Neerav Kingsland, Nelson Elhage, Nicholas Joseph, Noemí Mercado, Nova DasSarma, Oliver Rausch, Robin Larson, Sam McCandlish, Scott Johnston, Shauna Kravec, Sheer El Showk, Tamera Lanham, Timothy Telleen-Lawton, Tom Brown, Tom Henighan, Tristan Hume, Yuntao Bai, Zac Hatfield-Dodds, Jack Clark, Samuel R. Bowman, Amanda Askell, Roger Grosse, Danny Hernandez, Deep Ganguli, Evan Hubinger, Nicholas Schiefer, Jared Kaplan Anthropic, and A. I. Surge. Discovering language model behaviors with model-written evaluations. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 13387–13434, 12 2022.

[12] Francis Rhys Ward, Matt MacDermott, Francesco Belardinelli, Francesca Toni, and Tom Everitt. The reasons that agents act: Intention and in-strumental goals. *IFAAMAS*, 12, 2024.

[13] Francis Rhys Ward, Francesco Belardinelli, Francesca Toni, and Tom Everitt. Honesty is the best policy: Defining and mitigating ai deception. 12 2023.

[14] Rohin Shah, Vikrant Varma, Ramana Kumar, Mary Phuong, Victoria Krakovna, Jonathan Uesato, and Zac Kenton. Goal misgeneralization: Why correct specifications aren't enough for correct goals. 10 2022.