INSTITUTE OF TECHNOLOGY AND LEADERSHIP

COMPUTER SCIENCE

MELYSSA DE SOUZA GONZALES ROJAS

Project Proposal: Risk Analysis of Large Language Model Collapse Based on an Autonomous Agent

São Paulo

2025

MELYSSA DE SOUZA GONZALES ROJAS

Project Proposal: Risk Analysis of Large Language Model Collapse Based on an Autonomous Agent

Project proposal along with a literature review presented to the Computer Science program at the Institute of Technology and Leadership as a partial requirement for the first term grade.

Advisor: Prof. Dr. Rafael Will Macedo de Araujo

São Paulo
2025

ABSTRACT

This study investigates the impact of the interaction between autonomous agents and collapsed Large Language Models (LLMs), a phenomenon associated with the degradation of response quality due to the excessive use of synthetic data during training. The research analyzes the performance of AgentForce-based agents combined with the security mechanisms of the Einstein Trust Layer in dialogue scenarios involving collapsed models. Three fronts are explored: evaluation of the confidence of successor models when exposed to original data distributions; measurement of the guardrails' actions in filtering or transforming inappropriate responses; and analysis of the average toxicity of generated answers. The main contribution lies in providing practical insights for the development of more robust conversational systems, even in the face of progressive model collapse.

Keywords: Large Language Models, model collapse, autonomous agents, synthetic data, conversational systems, AgentForce, Einstein Trust Layer, guardrails, response toxicity

SUMMARY

1 INTRODUCTION

With the aim of improving dialogue circles in interactions between humans and chatbots, Large Language Models (LLMs) have occasionally been used to support better generalization and conversational flow. However, these models still showed limitations in capturing the complexity involved in conversations aimed at achieving specific user goals. In response to this challenge, autonomous agents emerged, whose main feature is the ability to perform structured planning by breaking down the final goal into subtasks. This enables greater precision during the interaction stages, iterative feedback, and prolonged memory use until the objective is achieved [2, 3, 5, 7].

The introduction of these autonomous agents, applied as conversational agents, has significantly contributed to improving the efficiency of understanding user goals in complex environments [4, 5, 7]. Moreover, there has been an observed increase in the agents' ability to adapt their behavior based on the feedback received throughout the interaction [7]. These agents, however, continue to rely on language models to support intent interpretation, assist in planning, and generate responses [2].

Recently, it has been observed that newer versions of LLMs may present degradation in response quality, contrary to the expectation of continuous improvement. This phenomenon is associated with the increased use of synthetic data generated by predecessor models during the training of successor models. In a scenario where human origin data becomes less predominant, models begin to be trained on increasingly synthesized distributions, which can result in narrowed response patterns, hallucination replication, and an overall decline in generation quality [1].

Given this context, the hypothesis arises that autonomous agents through their capacity for iterative planning, accurate task division, use of guardrails to mitigate hallucinations, and toxicity detection tools may attenuate the impacts of low-quality responses generated by collapsed models. The operation of these components may prevent inappropriate content from reaching the end user or at least significantly mitigate its impact.

This research focuses on analyzing the effects of the interaction between conversational agents and collapsed models in dialogue scenarios. The analysis is carried out across three main fronts: the first observes the behavior of successor models trained on synthesized distributions, evaluating their reaction to an original distribution and measuring the confidence assigned to high and low probability events; the second involves quantifying the responses generated by collapsed models that are filtered, transformed, or identified as inappropriate by the agents' guardrails, testing different configurations of these components; and the third concerns the analysis of the average toxicity of responses, based on the measurement of individual toxicity in specific scenarios.

This is a case study focused on analyzing the use of Salesforce's AgentForce autonomous agent technology, combined with the security mechanisms of the Einstein Trust Layer, to investigate the mitigation capacity of the effects caused by an external collapsed model [9, 10]. The goal is to assess the effectiveness of these technologies in maintaining interaction integrity and guiding users toward achieving their goals, even in the presence of a degraded model.

The main contribution of this work is to fill the existing gap in analyzing the impacts caused by the interaction between autonomous agents and collapsed LLMs, providing practical insights for the development of more robust and secure systems in AI-assisted conversational contexts.

2 RESEARCH METHODOLOGY

The research was conducted through a systematic literature review, with the primary objective of exploring topics related to collapsed models and autonomous agents, with a special focus on the theoretical enrichment of these technologies. It also aimed to investigate possible applications of this interaction, particularly in the context of LLM-based agents in corporate environments, in user support-related activities.

Given that these technologies are relatively recent, some limitations emerged during the process of searching and collecting articles, which made it difficult to gather a considerable amount of material. Moreover, the scarcity of publications on autonomous agents and collapsed models revealed a significant gap in the literature, which reinforces the relevance and originality of the proposed research.

The methodological process included detailed notetaking of the reviewed articles, followed by a pre-selection of the most relevant studies, which were considered essential for the development of this project. The article selection criteria were primarily based on two aspects: papers with a stronger theoretical focus, allowing for an in-depth analysis of the concepts and foundations of these technologies, and papers that highlighted use cases related to chat-based applications in corporate environments, as this is the core application area of the research. The main research sources included high-impact academic journals such as IEEE and Nature, as well as specialized technology and innovation websites. Technical documentation from Salesforce was also used to support the construction of the technical proposal, especially regarding the implementation of autonomous agents.

Most of the articles analyzed presented theoretical content, but it was also possible to identify a positive contradiction among the studies: while some suggest that autonomous agents help minimize the generation of inadequate responses by the models, others indicate that their use may worsen the problem, generating more erroneous or less effective responses. This disagreement among studies highlights the complexity of the topic and the need for further investigation into the interaction between autonomous agents and collapsed models, especially in corporate contexts.

This critical analysis, combined with the theoretical survey, was essential to identify gaps in the existing literature and support the construction of the technical proposal for this project.

3 RELATED WORK

This section discusses the concepts of the main technologies used in this research, based on a review of relevant articles on the topic.

3.1 Model Collapse

Model collapse is a phenomenon observed in generative artificial intelligence (AI) models, where the quality of the generated responses deteriorates over successive generations. This happens because, as new models are trained on data generated by previous versions, the presence of information originating directly from human interactions significantly decreases. As a result, the model loses diversity and becomes less capable of producing relevant and accurate responses, leading to increasingly significant limitations [1].

3.1.1 Stages of Model Collapse

The degeneration caused by model collapse can be understood in two stages: the initial stage and the advanced stage. In the beginning, the model starts to lose the tails of the data distribution, meaning it no longer accounts for rare or underrepresented information. As generations progress, it gradually diverges from the original distribution, reducing variability and focusing on more frequent patterns or responses [1].

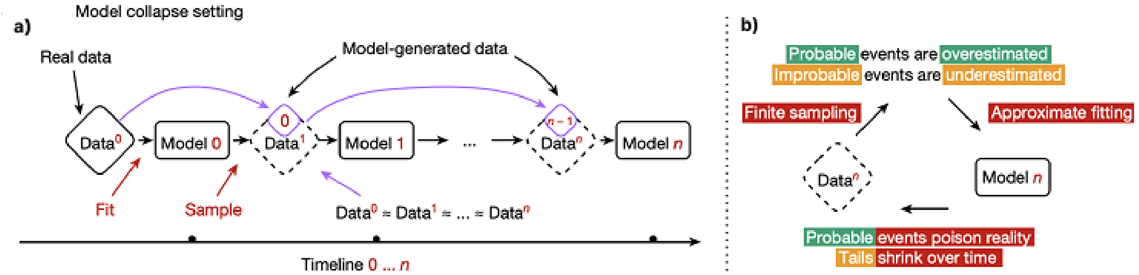

Figure 1: a. Flow of model collapse and the generation of its data distributions, focusing on model training based on the current data distribution, followed by the regrouping of a new dataset. b. Cycle of model reinforcement by high-probability events, leading to a worsening of reality in the data. Conversely, undervaluation of low-probability events and shrinkage of the data distribution tails [1].

In Figure 1a, the process begins with the "Data" from generation "0," consisting solely of content generated by humans. This data is used to train the "Model" of generation 0. Next, the subsequent model is trained with the data from the first generation, which has already been altered by being generated based on the previous model. This cycle repeats, creating new distributions successively. In Figure 1b, it is observed that, over the cycles, the most probable events are overestimated, while rare events are forgotten. As the number of samples is finite, the process eventually leads to model collapse. Over time, the models begin to reflect a distorted reality, where the tails of the distributions disappear, compromising the diversity and richness of the responses [1].

3.1.2 Perplexity of Model Collapse

Perplexity is a widely adopted metric for evaluating the performance of language models, reflecting the model's level of uncertainty when predicting the next word in a text sequence. Intuitively, perplexity can be understood as the average number of words among which the model "hesitates" at each generation step. The lower the perplexity, the higher the model's confidence in its predictions:

$$\text{Perplexidade} = \exp\left(-\frac{1}{N}\sum_{i=1}^{N}\log P(w_i \mid w_1, w_2, \ldots, w_{i-1})\right) \tag{1}$$

as defined in [8], metric is calculated based on the probabilities assigned to each word w_i in the sequence. For a sequence of N words, perplexity takes into account the model's "surprise" level regarding the occurrence of each word, given the previous context. For example, a perplexity of 5 indicates that, on average, the model considers five words to be equally probable at each step.

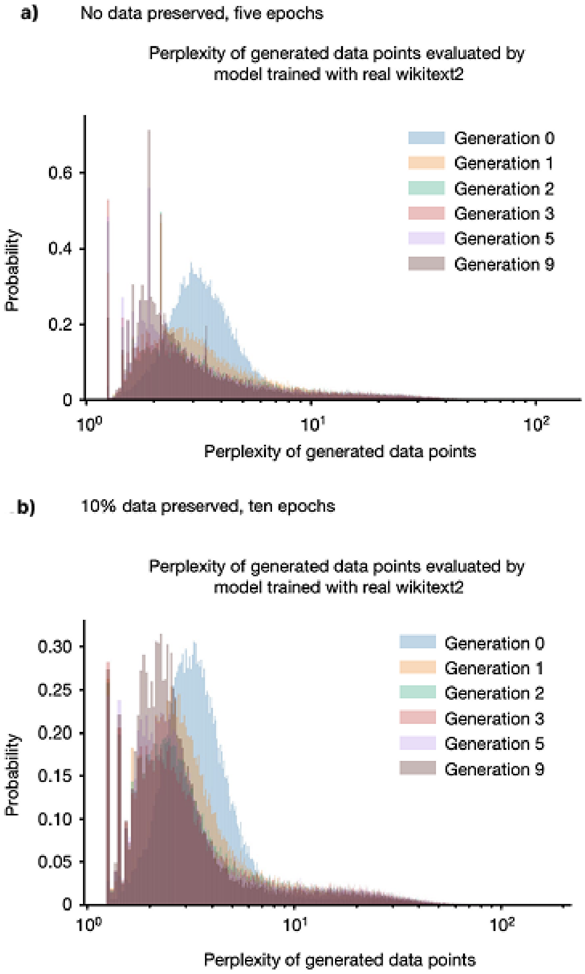

Figure 2: a,b Perplexity evaluation of different generations using a model trained on the original data distribution. Two approaches are shown: one with no preservation of the original distribution and another with 10% preservation. In both cases, model collapse still occurs [1].

In Figure 2a,b, the variation of perplexity across nine generations of models is illustrated, using the first generation, trained exclusively with human-curated data, as a comparative baseline. As the generations progress, it is observed that the new distributions shift to the left on the histogram and become narrower, indicating an increasing absorption of high-confidence patterns inherited from previous models. However, this trend reduces the models' ability to handle rare events. It is shown that the more recent generations have tails more concentrated to the right, revealing higher perplexity in these cases. This highlights that, with collapse, the models tend to ignore less frequent token sequences, becoming more prone to errors in less common situations. However, in the face of common events, the models appear more confident, although with reduced diversity in the generated responses [1].

3.1.3 Primary Cause of Model Degeneration

The main factor leading to model collapse is statistical approximation error. This error occurs due to the limitation imposed by the use of finite samples, meaning the model does not have access to the entire universe of possible data. In theory, this error would tend to decrease as the number of samples approaches infinity [1]. However, in practice, it arises because some information is lost during the successive stages of data regrouping, especially when there is an excessive concentration on data generated by predecessor models, to the detriment of original data from human interaction.

3.1.4 Mathematical Foundations of Data Distribution

When we refer to the i-th generation of a model, we are talking about a training cycle in which the model is adjusted based on a specific dataset, D_i. The data distribution associated with this generation is represented by P_i, indicating how the data is organized and what patterns the model learns from it [1].

The distribution of the next generation, P_{i+1}, is estimated based on the previous distribution, P_i, through an approximation function $F_\theta(P_i)$, which seeks to predict what the data distribution will be in the next generation. The new dataset, D_{i+1}, is constructed based on the following weighted combination:

$$P_{i+1} = \alpha_i P_{\theta_{i+1}} + \beta_i P_i + \gamma_i P_0 \tag{2}$$

in this equation [1], $P_{\theta_{i+1}}$ represents the data distribution generated by the new model, P_i corresponds to the data used in the previous generation, and P_0 refers to the original distribution of human-curated data. The coefficients α_i, β_i, and γ_i are non-negative and sum to 1, thus controlling the relative influence of each data source on the training of the new generation.

As generations progress, the model tends to train predominantly with data derived from its predecessors, reducing the diversity of the samples. The probability that certain states of the sample space are no longer represented increases. If the probability of a state occurring is q, the chance that this state is not sampled is given by $1 - q$. This means that as the model becomes more focused on frequent patterns, it gradually loses the ability to represent and generate responses associated with rare events [1].

This process leads to the convergence of the probability distribution to a delta function, a distribution where all the probability mass concentrates on a single point. In practical terms, the model starts to repeatedly generate the same response, losing its ability for diversity and generalization.

3.2 Agentic AI

Agentic AI, in the context of autonomous agents based on LLMs, is characterized by the ability to perform complex tasks autonomously, without the need for direct human supervision. These systems exhibit autonomy in managing their own resources and are capable of reasoning, adapting to different contexts, and making decisions based on circumstances in order to achieve their goals [2].

3.2.1 Agentic AI Architecture Types

The concept of Agentic AI encompasses different types of architectures, each representing a specific type of agent. Among these architectures, the following stand out: the Multi-Agent System (MAS), in which multiple autonomous agents collaborate or compete to achieve specific goals [2, 3]; the hierarchy based on reinforcement learning, where higher-level agents coordinate the actions of lower-level agents; and the goal-oriented modular architecture, where the agent performs functions divided into modular components, with each module responsible for a part of the task [2].

AgentForce aligns more closely with the goal-oriented modular architecture. In this approach, the agent is structured with specialized components, where each part of the system, such as the planner, topics, actions, guardrails, among others, has a well-defined function and operates in a modular way to achieve the user's final goal.

This organization allows for greater control, scalability, and traceability of the agent's actions [2].

3.2.2 Behavioral Dynamics in Multi-Agent Architectures

A representative architecture, used to exemplify the orchestration and components commonly employed in autonomous agents, can be observed in a scenario aimed at offering personalized recommendations to a user through chat interactions.

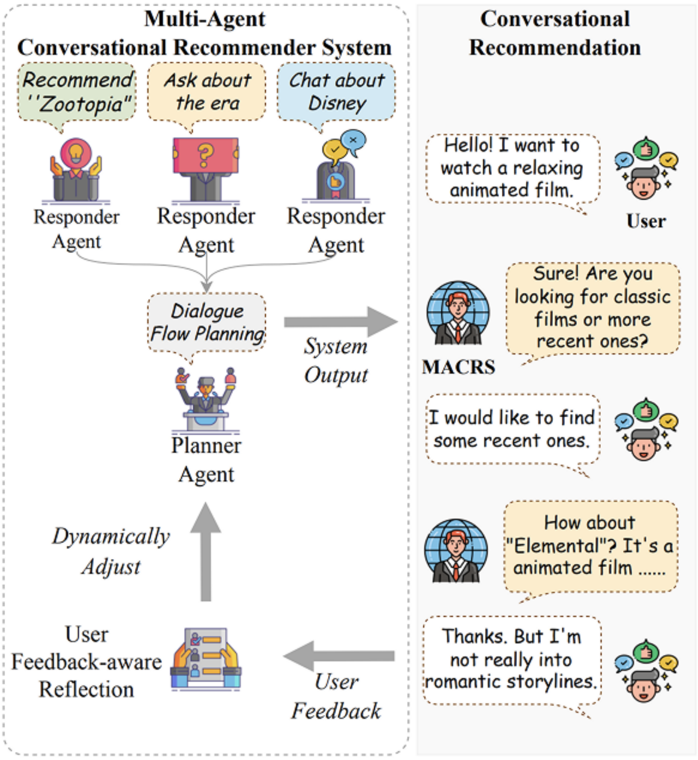
Multi-Agent Conversational Recommender System
Recommend "Zootopia"
Ask about the era
Chat about Disney
Responder Agent
Responder Agent
Responder Agent
Dialogue Flow Planning
System Output
Planner Agent
Dynamically Adjust
User Feedback-aware Reflection
User Feedback
Conversational Recommendation
Hello! I want to watch a relaxing animated film.
User
Sure! Are you looking for classic films or more recent ones?
MACRS
I would like to find some recent ones.
How about "Elemental"? It's a animated film ......
Thanks. But I'm not really into romantic storylines.

Figure 3: Example of modeling a Conversational Recommendation System (CRS) in a multi-agent environment. On the right, the dialogue between the user and the system; on the left, the collaboration between the planner and responder agents, with a reflection mechanism that guides and improves responses to better meet user needs [7].

In Figure 3, three main agents responsible for direct interactions with the user and a planner agent can be observed. Each agent has a specific function: the Questioning Agent formulates questions to identify the user's preferences; the Conversation Agent conducts casual dialogues, allowing the indirect revelation of preferences; and the Recommendation Agent generates suggestions based on previously collected information. These agents use memory, profile, and action modules, which enable them to learn from past interactions and generate more relevant responses [7].

The Planner Agent is responsible for coordinating the flow of the dialogue, selecting the best response from different options based on the conversation history, user context, and expected effectiveness of each response. To achieve this, it relies on both memory records and reflective strategies that allow dynamic adjustment of its approach [7].

During interactions, the system collects user feedback and uses it to continuously enhance the experience in two complementary ways. At the informational level, the feedback is analyzed to update the user profile, enabling more precise recommendations over time. At the strategic level, the system identifies failures in previous interactions and adjusts its dialogue strategies, aiming for more effective responses in future conversations based on accumulated learnings [7].

In summary, this multi-agent system operates as a collaborative framework in which each agent, with a well-defined function, contributes to a common goal, while the planner orchestrates decisions based on history, context, and refinement strategies [7]. This flow resembles the operation of a modular agent, where different internal modules perform specialized tasks and exchange information in a coordinated manner to achieve complex goals [2]. The main distinction between the two models lies in the functional distribution: in the modular agent, operations occur within a single entity composed of internal modules; in the multi-agent system, however, functions are distributed among distinct agents that cooperate with each other. Nevertheless, both models follow the

3.2.3 Transparency and Ethics in Agentic AI

The main security mechanisms associated with Agentic AI technology, responsible for ensuring compliance with ethical, legal, and transparency principles, involve both technical protocols and accountability strategies. Security protocols guide the system to follow objectives aligned with legal standards and ethical guidelines, while mechanisms such as fail-safe systems are designed to automatically interrupt potentially dangerous activities. Additionally, ethical guardrails are implemented, acting as behavioral barriers to ensure that the AI's decisions remain within socially acceptable standards, avoiding abuses and misuse [2].

Regarding transparency, the importance of trust and accountability in the application of Agentic AI systems is emphasized. In this context, audit trails can be used to document each logic and decision made by the system, allowing for subsequent review and critical analysis of the actions taken. Complementarily, self-documenting algorithms contribute to the system's explainability by generating detailed reports about its decisions during operation [2]. These resources reinforce transparency mechanisms and facilitate continuous feedback from users and developers.

3.2.4 Autonomous Agent for LLM Response Quality

When LLMs employed in interactive activities with users, face limitations when dealing with complex scenarios and maintaining a consistent self-definition during prolonged dialogues. Such limitations can undermine performance, especially in long-duration interactions. In this context, the implementation of autonomous agents, particularly through MAS architectures, has proven promising. These agents exhibit autonomy and are composed of integrated components focused on planning and goal setting, allowing for greater accuracy and maintaining focus during extended interactions with the user [4, 7].

Despite experimental advances that highlight the ability of these agents to mitigate LLM limitations in interactive tasks, significant contradictions arise. The use of autonomous agents in decentralized architectures can, at times, exacerbate problems associated with LLM-generated responses, due to the complexity inherent in decentralization, which makes centralized control of the system more difficult [6].

An example of this issue is observed in MAS systems where agents operate with high autonomy, seeking to maximize individual rewards associated with their own objectives. Such behavior can compromise the system's macro goal, especially if the interaction between agents is not strategically and collaboratively planned in advance, negatively affecting the overall efficiency of the application. Furthermore, if an agent or component of the architecture has vulnerabilities, whether due to malicious intent, configuration failures, or inefficiency, there is a risk of a cascading effect that compromises

3.3 AgentForce

The main objective of the project, as mentioned in the introduction, is to evaluate autonomous agents based on LLMs with collapse. To achieve this goal, the technology from AgentForce, developed by Salesforce, will be used. It provides essential components for building autonomous agents. Among these components, the highlights are the

Agent Builder (for constructing autonomous agents), the Model Builder (which allows the use of platform models or proprietary third-party models), and the Prompt Builder (which enables the creation of reusable prompt templates) [10].

The Agent Builder will be responsible for constructing the agent from a set of customizable components, allowing the connection with another essential tool: the Einstein Trust Layer, which ensures the implementation of security layers, guaranteeing privacy and ethical standards in the interactions performed by the agent [9]. The main components that make up the architecture of the Agent Builder include: Topics and Actions, the Reasoning Engine (named Atlas), and the LLM [10].

3.3.1 Topics and Actions

An agent contains a library of topics and actions, which are essential for its operation. Actions represent the tasks an agent can perform. For example, when a user requests help in drafting an email, the agent can initiate an action that drafts and reviews the email, using relevant data from Salesforce. Additionally, Salesforce provides some standard actions for common tasks, with the possibility of creating custom actions to address specific use cases for each company [10].

Topics are categories of actions, grouped according to a specific task to be performed. For example, a topic called Business Management may contain actions that help a sales representative stay organized, locate opportunities, find relevant contacts, create to-dos, and log calls. Within topics, actions function as the tools available to perform the work, while instructions guide how the actions should be carried out [10].

When an agent is triggered, or when a user asks a question or makes a request during a conversation, the agent compares the received request or task with the names and classification descriptions of the topics it is assigned to. The agent then classifies the request under the most appropriate topic, and based on the actions and instructions from the selected topic, it may execute one or more actions. If necessary, the agent can ask the user for additional information, such as a clarifying question or data needed to perform an action [10].

3.3.2 Reasoning Engine

The reasoning engine is responsible for coordinating how the agent triggers topics and actions during a conversation in order to fulfill the requested task. When the agent is triggered or when a user makes a question or request, the reasoning engine works alongside the LLM behind the scenes, performing the following functions: it interprets the user's trigger or request and classifies the request under a relevant topic; constructs iterative plans to achieve the requested goal; finds and executes the appropriate topics and actions to reach the goal [10].

3.3.3 Use of LLM within AgentForce

The agents use the power of an LLM to communicate with users and perform actions within the organization. The reasoning engine calls the LLM at different points during the execution of a task or interaction with the user. The number and frequency of LLM calls vary depending on the task to be performed and the topics and actions that are triggered. This continuous interaction with the LLM allows the agent to perform tasks intelligently and effectively, based on the analysis of data and the context provided during the interaction [10].

These components operate in an integrated manner to create highly functional autonomous agents, capable of performing complex tasks and interacting efficiently with users [10]. The architecture of AgentForce, by utilizing topics, actions, and an advanced reasoning engine, enables agents to not only perform automated tasks but also adapt their responses and actions according to the context and needs of each interaction.

3.3.4 Einstein Trust Layer and AI Guardrails

The Einstein Trust Layer is a set of resources, processes, and policies designed to ensure data privacy, improve AI accuracy, and promote responsibility in AI usage within the Salesforce system. In this research, it plays a key role by providing security layers during the interaction between the agent and the LLM [9].

Given the versatility of the layers present in this technology, two components stand out: the Toxicity Score and the Audit Trail. The first is a tool that measures the toxicity of a response generated by a model, with the aim of identifying messages that may disrespect the user. This tool is valuable for gaining insights into the model's performance [9].

The Audit Trail, on the other hand, is a tool that provides an overview of all interactions between user prompts, responses generated by the model from the management of the autonomous agent's planner, and the performance of instructions given to the conversational agent's topics. An important point to highlight is its ability to generate feedback aimed at improving the customization definitions of the agent's components [9].

Finally, AgentForce uses the ethical AI Guardrails to minimize hallucinations and ensure protection against malicious threats and attacks. This component can be customized to define its settings, using natural language to establish its instructions and behaviors, allowing it to set barriers for different types of messages generated by the LLM [11].

4 TECHNICAL PROPOSAL AND METHODOLOGICAL APPROACH

This research aims to analyze the behavior of LLMs subjected to collapse, in both early stage and late stage, when used in conjunction with autonomous agents. To achieve this, two collapsed models will be constructed and evaluated based on the perplexity metric, followed by experimentation in an autonomous agent implemented through the Agent Builder. As a comparative element, a model with data distributions originating from human authorship will also be incorporated.

The interaction between the autonomous agent and the LLMs will be evaluated through simulated use cases, consisting of dialogue rounds adapted to different scenarios. The generated responses will be quantified using two mathematical equations based on the performance of security components, with a focus on the Toxicity Score and Guardrails indicators.

4.1 Materials and Methods

For the execution of the experiments, the OPT-125m language model, provided by Meta on the Hugging Face platform, will be selected. This model was chosen for its pre-trained nature, which will allow time savings in the generation of new models from different fine-tuning iterations. The dataset to be used for fine-tuning will be wikitext2, which, with its general-domain nature, offers flexibility in adapting to distinct use cases and performance testing [1].

The construction of the synthetic dataset will be carried out from the original dataset using a sequence prediction-based approach. Initially, the corpus will be segmented into 64-token blocks. For each block, the model will be instructed to predict the next 64 tokens, resulting in the generation of new artificial sequences. This procedure will be applied to the entire original dataset and to the subsequent sets of future generations, aiming to produce a new dataset of equivalent size, entirely composed of data generated by the current model [1].

The experimental process will involve the creation of three model generations: the first will correspond to the original model fine-tuned with real data, while the subsequent two will be generated from synthetic data obtained as described above. This approach will allow the observation of different collapse levels across generations.

Regarding the autonomous agent, its implementation will be carried out through the Agent Builder tool. The topics and actions that make up the agent's behavior can either be native to the platform or customized according to the experiment's needs. Additionally, configurable guardrails will be used based on pre-established guidelines aimed at mitigating ethical and behavioral risks. The interaction with users will be implemented in a specific platform channel, and the orchestration of the agent's decisions will be managed by the Reasoning Engine. The language model used in interactions with the agent will be the collapsed LLM, as described in the generations.

4.2 Metrics and Evaluation

The evaluation of model collapse across generations will be conducted based on the perplexity metric (1). This metric allows for quantifying the degree of uncertainty in the model when predicting the next textual unit, being sensitive to both high-probability events and rare events. The expectation is that models in advanced stages of collapse will present higher perplexities, especially in low-frequency events, indicating degradation in generation capability and a greater tendency to produce generic patterns and incoherent responses.

In the context of the autonomous agent's interaction with collapsed models, an evaluation will be conducted focusing on the security components of guardrails, where N indicates the number of dialogue rounds and c the current configuration of the agent. To this end, the Guardrail Action Index (GAI) is defined, which quantifies the proportion of responses that were blocked or modified by the security mechanisms in relation to the total number of responses generated during the simulations:

$$\mathrm{GAI}_c = \frac{\sum_{i=1}^{N} \left(R_b^{(i)} + R_m^{(i)} \right)}{\sum_{i=1}^{N} R^{(i)}} \tag{3}$$

where $R^{(i)}$ represents the total number of responses generated in the i-th dialogue round, $R_b^{(i)}$ corresponds to the number of responses blocked, and $R_m^{(i)}$ to those that were modified by the guardrails. The value of GAI_c ranges between 0 and 1, with values closer to 1 indicating a greater preventive action of the agent against inappropriate responses generated by collapsed models.

Additionally, the quality of the generated responses will be evaluated based on their average toxicity. To this end, the General Toxicity (GT) equation is defined, which measures the degree of toxicity of the agent's interactions:

$$\mathrm{GT}_c = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{1}{R^{(i)}} \sum_{j=1}^{R^{(i)}} t_{ij} \right) \tag{4}$$

where t_{ij} represents the toxicity index of the j-th response in the i-th dialogue round, with values between 0 and 1. Higher values of GT_c, in this case, closer to 1, indicate greater toxicity on the part of the collapsed model in generating text, meaning potentially offensive responses.

5 CONCLUSION

It is expected that by the end of this project, a deeper understanding of the interaction between collapsed language models and autonomous agents will be achieved, especially evaluating the impact of different agent configurations on the overall system performance. The goal is also to contribute to the advancement of knowledge about the collapse phenomenon in LLMs, particularly in contexts of integration with agents, filling an existing gap in the literature regarding practical experimentation with conversational agents.

The next steps include the construction of the three planned models, the definition of the configurations for the autonomous agents, and subsequently, the execution of simulations with test scenarios composed of dialogue rounds in a chat environment.

Although the collapsed model is recognized as a potentially inevitable phenomenon, this research aims to investigate mitigation strategies and best practices that can avoid or minimize its effects. As identified in the literature, the use of autonomous agents in conjunction with collapsed models can represent both an opportunity and a risk, depending on how this integration is conducted. Finally, it is emphasized that models trained from a balanced distribution of synthetic and human-authored data tend to have better generalization capabilities, which can result in better performance in both high and low probability events [1]. Thus, ensuring auditing and governance mechanisms throughout the lifecycle of these models can significantly contribute to the effectiveness of both the model and the agent utilizing it in response generation and interpretation tasks.

REFERENCES

[1] SHUMAILOV, Ilia et al. **AI models collapse when trained on recursively generated data**. Nature, v. 631, n. 8022, p. 755–759, 2024. Available at: https://www.nature.com/articles/s41586-024-07566-y. Accessed on: Feb. 10, 2025.

[2] ACHARYA, Deepak Bhaskar; KUPPAN, Karthigeyan; DIVYA, B. **Agentic AI: Autonomous Intelligence for Complex Goals–A Comprehensive Survey**. IEEE Access, 2025. Available at: https://ieeexplore.ieee.org/document/10849561. Accessed on: Feb. 10, 2025.

[3] CRUZ, Carlos Jose Xavier. **Transforming Competition into Collaboration: The Revolutionary Role of Multi-Agent Systems and Language Models in Modern Organizations**. arXiv preprint arXiv:2403.07769, 2024. Available at: https://arxiv.org/abs/2403.07769. Accessed on: Feb. 27, 2025.

[4] XU, Zhenjie et al. **Mitigating Social Bias in Large Language Models: A Multi-Objective Approach within a Multi-Agent Framework**. arXiv preprint arXiv:2412.15504, 2024. Available at: https://arxiv.org/abs/2412.15504. Accessed on: Feb. 22, 2025.

[5] THAKKAR, Param; YADAV, Anushka. **Personalized Recommendation Systems using Multimodal, Autonomous, Multi Agent Systems**. arXiv preprint arXiv:2410.19855, 2024. Available at: https://arxiv.org/abs/2410.19855. Accessed on: Feb. 22, 2025.

[6] RANJAN, Rajesh; GUPTA, Shailja; SINGH, Surya Narayan. **Fairness in Multi-Agent AI: A Unified Framework for Ethical and Equitable Autonomous Systems**. arXiv preprint arXiv:2502.07254, 2025. Available at: https://arxiv.org/abs/2502.07254. Accessed on: Feb. 22, 2025.

[7] FANG, Jiabao et al. **A multi-agent conversational recommender system**. arXiv preprint arXiv:2402.01135, 2024. Available at: https://arxiv.org/abs/2402.01135. Accessed on: Feb. 22, 2025.

[8] WU, Chao-Chung et al. **Clear Minds Think Alike: What Makes LLM Fine-tuning**

Robust? A Study of Token Perplexity. arXiv preprint arXiv:2501.14315, 2025. Available at: https://arxiv.org/abs/2501.14315. Accessed on: Apr. 6, 2025.

[9] SALESFORCE. **Einstein Trust Layer: Response Journey**. Salesforce, 2023. Technical Report. Available at: https://help.salesforce.com/s/articleView?id=ai.generative_ai_trust_arch2.htm&type=5. Accessed on: Apr. 5, 2025.

[10] SALESFORCE. **The Building Blocks of Agents**. Salesforce, 2025. Technical Report. Available at: https://help.salesforce.com/s/articleView?id=ai.copilot_building_blocks.htm&language=en_US&type=5. Accessed on: Apr. 5, 2025.

[11] SALESFORCE. **Trust and Agentforce**. Salesforce, 2025. Technical Report. Available at: https://help.salesforce.com/s/articleView?id=ai.copilot_trust.htm&type=5. Accessed on: Apr. 5, 2025.