SAGE: Safe, Adaptive Generation Engine for Long Form Document Generation in Collaborative, High Stakes Domains

Abrar Rahman
Benki

Anish Sundar
Benki

With
Goodfire and Apart Research

Abstract

Long-form document generation for high-stakes financial services—such as deal memos, IPO prospectuses, and compliance filings—requires synthesizing data-driven accuracy, strategic narrative, and collaborative feedback from diverse stakeholders. While large language models (LLMs) excel at short-form content, generating coherent long-form documents with multiple stakeholders remains a critical challenge, particularly in regulated industries due to their lack of interpretability.

We present SAGE (Secure Agentic Generative Editor), a framework for drafting, iterating, and achieving multi-party consensus for long-form documents. SAGE introduces three key innovations: (1) a tree-structured document representation with multi-agent control flow, (2) sparse autoencoder-based explainable feedback to maintain cross-document consistency, and (3) a version control mechanism that tracks document evolution and stakeholder contributions.

Keywords: *AI Observability, Mechanistic Interpretability, Model Reprogramming, Version Control, Agentic Frameworks, Computational Finance, Risk Management*

1. Introduction

This exploration began with the high-level vision of bridging the rapid innovation of open-source AI with Wall Street's need for trust, accountability, and safety.

The deployment of large language models (LLMs) in high-stakes domains presents a fundamental challenge: how do we maintain precise control and transparency over AI systems while leveraging their powerful generative capabilities? This question becomes particularly acute in the context of

Table 1: Sample Long Form Doc

```
├── Executive Summary
│   ├── Key Findings
│   └── Recommendations
├── Analysis
│   ├── Market Overview
│   └── Financial Projections
└── Appendices
    └── Data Sources
```

long-form document generation, where outputs must maintain coherence and accuracy across hundreds of pages while satisfying complex regulatory requirements and incorporating multiple stakeholders' inputs.

SAGE (Safe, Adaptive Generation Engine) introduces a novel approach to this challenge by combining mechanistic interpretability with collaborative control mechanisms. At its core, SAGE demonstrates that we can "open the black box" of language models through a structured decomposition of both the generation process and the resulting artifacts.

2. Code

GitHub: https://github.com/benki-finance/MemoMaker
Vercel: https://benki-memomaker.vercel.app

Myra, Eric, Tom, and Dan—thanks for betting on us early.

3. Overview

3.1 Long Form Document Structure Formalization

Let D be a document represented as a directed tree $T = (V, E, \lambda)$ where:

- V: Set of nodes (document sections)
- $E \subseteq V \times V$: Directed edges representing dependencies
- $\lambda: V \rightarrow L$: Labeling function mapping nodes to content types

Content types $L = \{l_1, ..., l_n\}$ may include: FINANCIALS, ex quarterly earnings reports, ERP/accounting data; MARKET: market trends, the first- and second- order Greeks; LEGAL: localization and internationalization, regulatory compliance; NARRATIVE: strategic storytelling; and HYBRID: mixed content types (ex: tables, graphs).

3.2 Node Properties

Each node $v \in V$ has associated properties:

```
Node := {
    id: UUID,
    type: L,
    content: String,
    dependencies: Set[UUID],
    sources: Map[UUID → Source],
    contributors: Map[UUID → Contributor],
```

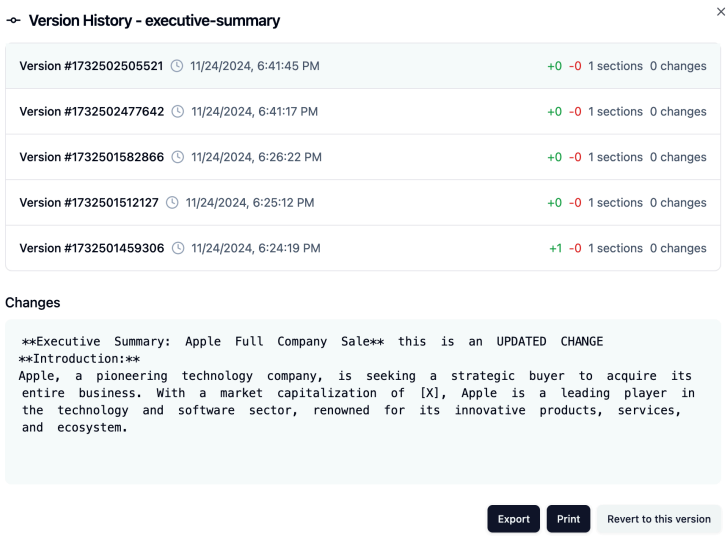
```
Contributor := {
    id: UUID,
    role: String,
    modifications: List[Modification],
    ownership_weight: Float[0,1]
}
```

```
    version_history: List[Version],
    steering_params: SteeringConfig,
    confidence: Float[0,1],
    last_modified: Timestamp
}
```

3.3 Version Control System

SAGE maintains document state as a directed acyclic graph (DAG). We can define an illustrative minimal version graph G = (S, T, μ) where:

- S: Set of document states
- T: Set of transitions between states
- μ: T → M: Mapping to modification metadata



For example, a merge operation in SAGE could be formalized as a single operation, a parallel modifications m_1, m_2 on node v:

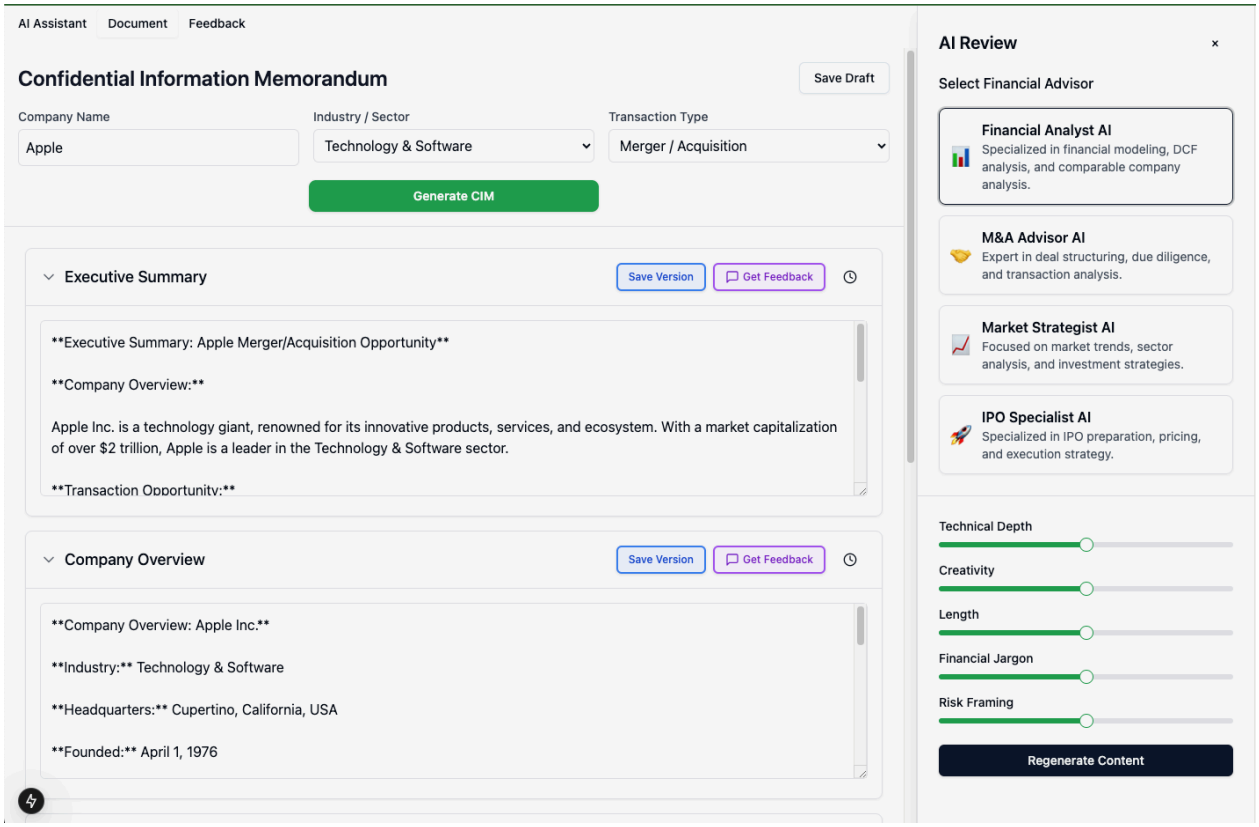
```
merge(v, m₁, m₂) := {
    content: resolve_conflicts(v.content, m₁, m₂),
    confidence: min(m₁.confidence, m₂.confidence),
    contributors: update_weights(v.contributors, m₁,
m₂),
    steering: combine_steering(m₁.steering, m₂.steering)
}
```

We intend to extend the SAGE framework to include analogues to other git commands, eventually building towards full pull requests, CI/CD pipelines, and more.

3.4 Collaborative Workflow Interface

Each node in SAGE supports instant expansion/collapse, context-aware feature controls, and version history visualization. We propose several modes of operation to demonstrate SAGE's capabilities: **Ideation**, **Review, and Update**.

The paradigm relies on the assumption that human beings still do most



3.5 Local Safety Metrics

SAGE is a living framework built with Explainable AI (XAI) in mind. Though we haven't yet implemented our vision to the fullest in regards to preference-tuned evaluations at scale, consider the following toy example. For each agent A_v, we define:

1. **Feature Alignment Score**: FA(A_v) = cos(f_v, f_target) where f_v is the agent's feature vector and f_target is the target alignment
2. **Safety Probability**: P(safe|A_v) = ∏ P(invariant_i|A_v) across all safety invariants i

3.2 Global Safety Guarantees

For document D with agents A = {A_1, ..., A_n}:

1. **Collaborative Coherence**: CC(D) = min_{i,j} cos(f_i, f_j) for all agent pairs (i,j)
2. **Global Safety Probability**: P(safe|D) = ∏ P(safe|A_v) across all agents v

3.3 Probabilistic Guarantees

Node-Level Safety

For a node v handled by agent A_v :

$P(\text{safe}(v)) \geq 1 - \epsilon$

where safety is an aggregate customer-preference specific theoretical measure whichencompasses compliance, alignment, and factual accuracy.

Document-Level Safety
For document D with nodes V :

$$P(\text{safe}(D)) = P\left(\bigwedge_{v \in V} \text{safe}(v)\right) \geq 1 - |V|\epsilon$$

4. Discussion and Conclusion

Combining the flexibility of Notion with the version control of GitHub, Benki enables deal teams to generate, refine, and collaborate on deal documents in real-time, with full visibility into every AI-driven change. SAGE is the first step on our journey.

The current state of AI in finance is a perfect storm of overhyped potential and lack of real-world applicability. The core issue is opacity—AI models in finance today are often black-box systems, producing content that looks polished but lacks sophistication and transparency. As banks and firms increasingly rely on AI to handle critical documents and workflows, they're blind to the risks of unrefined, overtrained outputs. This is eerily reminiscent of the 2008 financial crisis—an overconfidence in systems that weren't properly scrutinized, leading to massive regulatory and operational failures. Without a robust framework to ensure transparency, adaptability, and accountability, these firms are setting themselves up for a repeat of history.

Benki provides a pathway to scalable adoption of LLMs in financial services while preserving the intellectual rigor of deal-making processes. Some are building a dystopic swarm of agents that seek to replace us. Others believe in a utopic future where agentic workflows turn every employee into a 100x'er. Benki is a pragmatic solution for the middle ground; a novel paradigm to enable deal teams to maintain their unique voice, reduce compliance risk, and work side-by-side with AI via battle-tested analogues from other industries.

References

1. Elhage, N., et al. (2022). "A Mathematical Framework for Transformer Circuits." arXiv:2202.01110
2. Wei, J., et al. (2024). "Long-form Factuality in Large Language Models." Proceedings of ACL 2024
3. Sharkey, A., et al. (2022). "Sparse Autoencoders for Interpretable AI." NeurIPS 2022
4. Liang, Y., et al. (2024). "Planning in Long-form Text Generation." ICLR 2024
5. Beltagy, I., Peters, M. E., & Cohan, A. (2020). "Longformer: The Long-document Transformer." EMNLP 2020