# Agentic Multi-Document Summarization with Anchoring and Consolidation of Parallel Events: An Experiment with the Four Gospels

**André F. de C. C. Cunha**[1,2], **Carlos A. Sena**[1,2], **Jacson R. Barbosa**[1]

[1]AKCIT – Instituto de Informática – Universidade Federal de Goiás (UFG)

[2]POSITIVO Tecnologia

andrefccc@positivosmais.com, carlos.sena@positivo.com.br, jacson_rodrigues@ufg.br

***Abstract.*** *This study investigates multi-document summarization (MDS) of long narratives, a relevant challenge in Natural Language Processing (NLP) due to difficulties in semantic integration and preservation of temporal coherence across parallel sources, with applications in historical and documentary narratives. We specifically position this work within the sub-problem of Narrative Consolidation (NC), which extends classical MDS by prioritizing chronological integrity and informational completeness over compression. We develop and evaluate an agent-based MDS prototype with temporal anchoring and abstractive consolidation of parallel events, applied to the four Gospels in XML format. The system comprises a modular pipeline with four autonomous agents: event anchoring; abstractive consolidation; narrative organization; and narrative fluency. Implemented in Python using LangChain and LangGraph, with events modeled as temporal graphs. Validation using automatic metrics (Kendall's Tau, ROUGE-L, and BERTScore) and qualitative expert analysis indicates gains in semantic fidelity and narrative fluency, outperforming state-of-the-art LLMs in zero-shot regime (BERTScore: 0.9379 vs. 0.9013 for the best baseline). The results demonstrate the technical feasibility of agentic orchestration as a reliable approach for multi-document consolidation.*
*Keywords: Multi-Document Summarization; Autonomous Agents; Large Language Models; Temporal Anchoring; Natural Language Processing.*

***Resumo.*** *Esta pesquisa investiga a sumarização multidocumento (MDS) de narrativas longas, problema relevante do Processamento de Linguagem Natural (PLN) devido às dificuldades de integração semântica e preservação da coerência temporal entre fontes paralelas, com aplicações em narrativas históricas e documentais. Posicionamo-nos especificamente no sub-problema de Consolidação Narrativa (NC), que estende a MDS clássica ao priorizar integridade cronológica e completude informacional em vez de compressão. Desenvolvemos e avaliamos um protótipo agentizado de MDS com ancoragem temporal e consolidação abstrativa de eventos paralelos, aplicado aos quatro Evangelhos em formato XML. O sistema compreende uma pipeline modular com quatro agentes autônomos: ancoragem de eventos; consolidação abstrativa; organização de narrativa e fluência narrativa. Implementados em Python com LangChain e LangGraph, modelando eventos como grafos temporais. A validação por métricas automáticas (Kendall's Tau, ROUGE-L e*

*BERTScore) e análise qualitativa por especialista indica ganhos em fidelidade semântica e fluidez narrativa, superando LLMs de estado da arte em regime zero-shot (BERTScore: 0,9379 vs. 0,9013 do melhor baseline). Os resultados demonstram a viabilidade técnica da orquestração agentizada como abordagem confiável para consolidação multidocumento.*
*Palavras-chave: Sumarização Multidocumento; Agentes Autônomos; Grandes Modelos de Linguagem; Ancoragem Temporal; Processamento de Linguagem Natural.*

## 1. Introdução

A Sumarização Multidocumento (MDS, do inglês *Multi-Document Summarization*) constitui um dos desafios centrais do PLN, especialmente quando aplicada a narrativas longas e paralelas que descrevem eventos semelhantes sob perspectivas distintas. Embora modelos neurais contemporâneos como PEGASUS [Zhang et al. 2020a] e PRIMERA [Xiao et al. 2022] tenham alcançado avanços significativos na sumarização de documentos únicos, persistem limitações na preservação da coerência temporal, na integração semântica entre fontes e na redução adequada de redundâncias em contextos multidocumento [Ma et al. 2023].

Trabalhos recentes argumentam que a unificação de narrativas multi-perspectiva constitui uma tarefa distinta da sumarização, com ênfase em integridade cronológica e completude em vez de compressão [Finger et al. 2025].

A complexidade desse problema é acentuada em corpora narrativos e históricos, nos quais a ordem dos eventos e a coexistência de versões paralelas são elementos centrais para a compreensão global do conteúdo. A ausência de mecanismos explícitos de ancoragem e alinhamento de eventos dificulta a reconstrução de narrativas unificadas e compromete a fidelidade factual do resumo gerado.

Historicamente, tentativas de unificar versões narrativas diversas remontam à Antiguidade. Um exemplo notável é o *Diatessaron*, de Taciano (século II), que buscou harmonizar os quatro Evangelhos (Mateus, Marcos, Lucas e João) em um relato único e coerente [Petersen 1994]. Essa tarefa permanece complexa até hoje: requer identificar eventos equivalentes, resolver divergências e preservar detalhes únicos de cada fonte.

Diante desse cenário, este trabalho tem por objetivo desenvolver e avaliar um protótipo agentizado de Consolidação Narrativa (NC, do inglês *Narrative Consolidation*) [Finger et al. 2025]com ancoragem temporal e consolidação abstrativa de eventos paralelos, aplicado aos quatro Evangelhos em formato XML, visando à geração de narrativas unificadas mais coerentes, completas e cronologicamente consistentes.

## 2. Trabalhos Relacionados

A MDS tem sido abordada por métodos extrativos e abstrativos. Barzilay et al. [Barzilay et al. 1999] investigaram pioneiramente a fusão informacional em MDS, e o LexRank [Erkan and Radev 2004] propôs centralidade grafal para seleção de sentenças representativas. Ma et al. [Ma et al. 2023] sistematizam avanços com aprendizado profundo, destacando limitações em coerência temporal e integração de narrativas paralelas. Santana et al. [Santana et al. 2023] oferecem uma revisão ampla sobre extração de nar-

rativas a partir de texto, área diretamente relacionada ao problema de consolidação multidocumento. Yasunaga et al. [Yasunaga et al. 2017] aplicaram grafos neurais à MDS, e PRIMERA [Xiao et al. 2022] introduziu pré-treinamento específico para múltiplos documentos.

Na vertente temporal, Bedi et al. [Bedi et al. 2017] exploraram a geração de linhas do tempo a partir de textos históricos. Mamidala e Sanampudi [Mamidala and Sanampudi 2021] propuseram um *framework* específico para MDS temporal (MDTS). Yu et al. [Yu et al. 2021] investigaram a geração de múltiplas linhas do tempo (MTLS). Song et al. [Song et al. 2025] demonstraram a relevância de raciocínio temporal explícito para sumarização de linhas do tempo em mídias sociais.

No domínio de agentes e LLMs, Wooldridge [Wooldridge 2009] e Russell e Norvig [Russell and Norvig 2021] fundamentam sistemas multiagênticos e tomada de decisão autônoma. *Frameworks* como LangChain e LangGraph [LangChain Team 2024] viabilizam a orquestração de agentes como grafos de estado, permitindo controle de fluxo, rastreabilidade e avaliação isolada de cada etapa do processo.

Mais diretamente relacionado, Finger et al. [Finger et al. 2025] formalizam a consolidação de narrativas como tarefa distinta da MDS e propõem o *Temporal Alignment Event Graph* (TAEG), abordagem extrativa que, a partir de uma linha do tempo canônica fornecida como entrada, garante a ordenação temporal por construção sobre o mesmo corpus dos quatro Evangelhos.

O presente trabalho se diferencia ao adotar uma estratégia abstrativa e agentizada integrando ancoragem semântica de eventos, consolidação abstrativa incremental e organização cronológica em uma pipeline avaliada quantitativa e qualitativamente sobre um corpus narrativo paralelo de longa extensão, na qual as fontes são fundidas por modelos de linguagem e o posicionamento de eventos exclusivos é inferido em tempo de execução, sem depender de uma linha do tempo externa.

## 3. Descrição da Tecnologia

A tecnologia proposta consiste em um protótipo agentizado de NC cujo propósito é alinhar, consolidar e enriquecer eventos paralelos descritos em múltiplos documentos narrativos longos. O corpus experimental compreende os relatos da Semana Santa nos quatro Evangelhos em formato XML, escolhido por concentrar eventos paralelos bem delimitados, narrados com variações significativas de ordem, ênfase e detalhamento.

### 3.1. Especificações Técnicas

O sistema foi operacionalizado por meio de: (i) **Orquestração (LangGraph)**, que define e controla o fluxo de execução como um grafo de estados (*StateGraph*), garantindo previsibilidade e extensibilidade arquitetural; (ii) **Núcleo Cognitivo (GPT-4o)**, modelo responsável pelo julgamento semântico, consolidação abstrativa e geração textual fluente, instanciado com *system prompts* especializados por agente; e (iii) **Linguagem Python**, com monitoramento via LangFuse para rastreamento detalhado das interações, decisões intermediárias e latência de cada etapa.

### 3.2. Arquitetura dos Agentes

A pipeline é composta por quatro agentes que operam em sequência sobre um estado compartilhado, ilustrados na Figura 1. Os três primeiros concentram-se na recuperação, fusão

e ordenação dos eventos; o último cuida do acabamento textual. A decisão de decompor o problema em agentes especializados, em vez de empregar um único modelo *end-to-end*, decorre da natureza heterogênea das decisões envolvidas sejam elas semânticas, temporais ou discursivas. A decisão também se baseia no interesse em avaliar isoladamente cada etapa do processo [Wooldridge 2009].

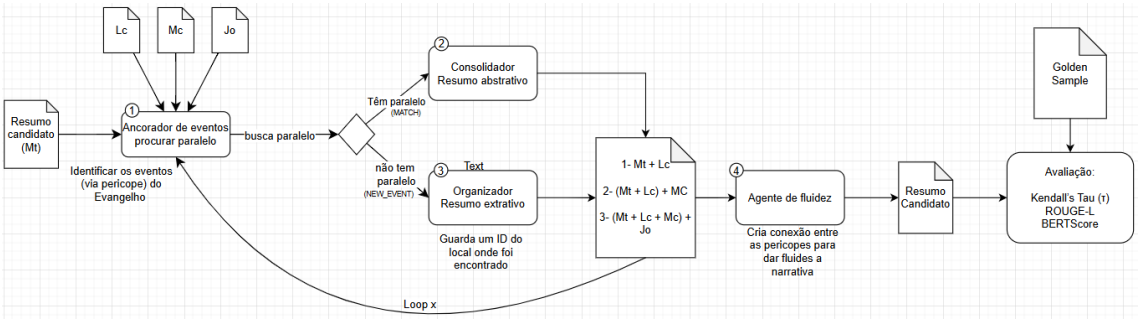


**Figura 1. Pipeline dos agentes do sistema.**

### 3.2.1. Agente Ancorador de Eventos

O primeiro agente decide, para cada perícope de um evangelho secundário, se ela descreve um evento já registrado a partir do evangelho-base (Mateus) ou um acontecimento inédito. Ele recebe quatro entradas: a perícope-alvo de Mateus, que serve de âncora cronológica; o conjunto de perícopes do evangelho candidato, que funciona como espaço de busca; os metadados estruturais de livro, capítulo e versículo; e o registro de eventos já processados, mantido no estado compartilhado para evitar reprocessamento.

A decisão ocorre em dois estágios. No primeiro, calcula-se a similaridade de cosseno entre a perícope-alvo e cada candidata, selecionando apenas a de maior pontuação, desde que ultrapasse um limiar de 0,60, valor definido empiricamente para reduzir falsos positivos sem sacrificar a cobertura. A escolha da representação vetorial foi precedida de uma comparação entre TF-IDF, similaridade de Levenshtein, índice de Jaccard e embeddings densos da família Gemma, executados localmente; os embeddings densos apresentaram o melhor equilíbrio entre custo computacional e capacidade de discriminação para perícopes longas, e por isso foram adotados nessa filtragem.

No segundo estágio, o candidato pré-selecionado é submetido ao GPT-4o, instanciado com um *prompt* de sistema que orienta o julgamento de equivalência pela correspondência da ação central e das figuras envolvidas, tolerando variações de estilo, terminologia e ordem narrativa. O modelo devolve um JSON com o veredito (MATCH ou NEW_EVENT), um grau de confiança no intervalo $[0, 1]$ e uma justificativa textual. O agente encapsula essa resposta em um objeto `TemporalAnchoringResult`, que preserva a perícope-alvo e a candidata para fins de auditoria e, no caso de correspondência, agrupa ambas em um *cluster* pronto para consumo pelo agente seguinte.

Esse agente funciona como o roteador do sistema. Diante de um MATCH, aciona imediatamente a consolidação e registra o identificador da candidata em `processed_candidate_ids`, garantindo deduplicação estrita. Diante de um NEW_EVENT, encaminha a perícope a um *buffer* de novos eventos e associa a ela um

`anchor_id` que aponta para o evento de Mateus em análise no momento, informação que o agente organizador usará depois para posicionar o evento exclusivo na narrativa.

### 3.2.2. Agente de Consolidação Abstrativa

Quando dois relatos do mesmo episódio são reconhecidos como equivalentes, cabe ao segundo agente fundi-los em um único texto. A operação é incremental: a cada nova correspondência, a versão consolidada do evento substitui a anterior no estado compartilhado, de modo que um episódio narrado pelos quatro evangelhos seja enriquecido em quatro passagens sucessivas pelo agente. As entradas são a perícope-base, que carrega o estado acumulado, e a perícope recém-validada, tratada como incremento; delas o agente extrai título e texto, injetados no *prompt* do modelo.

O *prompt* de consolidação instrui a produzir um texto maximalista, isto é, a incorporar tudo o que cada fonte traz de distinto. Restrições proíbem a supressão de nomes próprios, locais, personagens, ações características ou citações diretas, de forma que a abstração se limite à organização da narrativa e não implique perda de conteúdo factual. A saída do modelo (título harmonizado e texto unificado) é encapsulada em um objeto `ConsolidationResult` e validada antes de retornar ao estado compartilhado.

É esse agente que determina a densidade informacional da narrativa. Ao converter a sobreposição entre os evangelhos em enriquecimento progressivo, em vez de simples justaposição, ele busca uma saída ao mesmo tempo completa, não redundante e factualmente íntegra.

### 3.2.3. Agente Organizador de Narrativa

O terceiro agente não recorre a modelos de linguagem: trata-se de um algoritmo determinístico responsável por montar a linha do tempo definitiva. Ele consome três estruturas: a linha consolidada, que preserva a ordem cronológica de Mateus; o *buffer* de eventos ancorados, identificados em tempo de execução e portadores de um `anchor_id`; e o *buffer* de eventos órfãos, coletados em uma varredura final e desprovidos de vínculo posicional.

A montagem percorre a linha consolidada e insere eventos ancorados logo após seu evento de referência, preservando o contexto local. Ao término, os eventos órfãos são anexados ao final da narrativa, opção deliberada para evitar ordenações especulativas e alucinações temporais. O resultado é uma lista linearizada que reúne eventos consolidados, ancorados e órfãos, pronta para a renderização textual.

A separação entre as etapas de recuperação e ordenação, detalhada na trajetória de desenvolvimento (Seção 4.2), privilegia a completude: nenhum evento é perdido, ainda que alguns sejam posicionados de maneira menos precisa. Essa simplificação é assumida como limitação, a estratégia garante cobertura total, mas não a inserção cronologicamente ótima de cada evento, com reflexo direto na coerência temporal medida adiante.

### 3.2.4. Agente de Fluência e Coesão Narrativa

O último agente recompõe a sequência de eventos discretos em um texto corrido, eliminando a impressão de "colcha de retalhos" típica de saídas geradas por etapas, sem

alterar o conteúdo já consolidado. Para não exceder a janela de contexto nem encarecer a execução, o agente não submete a narrativa inteira ao modelo: ele monta pares locais de transição, formados pelos últimos 300 caracteres de um evento, pelos primeiros 300 do evento seguinte e pelos respectivos títulos.

A atuação do modelo é estritamente limitada por uma regra de imutabilidade, que o proíbe de reescrever, resumir ou expandir qualquer trecho. A única saída permitida é a sugestão de um conector discursivo entre dois eventos, escolhido entre quatro categorias: temporal ("no dia seguinte"), adversativa ("entretanto"), causal ("por causa disso") ou nula, quando a transição já flui naturalmente e nenhum conector é inserido. Essa parcimônia evita o excesso de marcadores artificiais.

O texto final é montado por um procedimento determinístico que intercala os textos originais, mantidos integralmente, com os conectores aprovados. Uma verificação de integridade baseada em busca exata de substrings confirma que a totalidade do conteúdo de entrada está presente na saída, garantindo formalmente que o modelo não removeu nem inventou informação. Enquanto os agentes anteriores cuidam de precisão semântica, cobertura e ordenação, este responde pela legibilidade do produto final.

### 3.3. Design do Fluxo de Dados

O sistema utiliza um **Grafo de Estado Compartilhado** (*Shared State Graph*), no qual os agentes leem e escrevem em um objeto de estado comum que trafega pela *pipeline*. O fluxo compreende: (1) ingestão de arquivos XML e conversão em objetos de domínio fortemente tipados; (2) processamento nodal sequencial pelos quatro agentes; e (3) geração de dois artefatos de saída, texto narrativo fluido e relatório estruturado, em JSON com metadados de rastreabilidade por sentença. O diagrama de fluxo de orquestração dos agentes é apresentado na Figura 2.

## 4. Procedimentos Metodológicos

### 4.1. Corpus e *Golden Sample*

O processamento foi restrito aos relatos da Semana Santa nos quatro Evangelhos, na versão em inglês *New International Version* (NIV) [Biblica, Inc. 2011] (35% do conteúdo total dos textos), previamente estruturados em formato XML padronizado com marcações explícitas de livro, capítulo, versículo e perícopes.A avaliação quantitativa requereu um *golden sample*, construído a partir de um resumo extrativo ajustado manualmente para coerência cronológica e completude informacional [Cunha 2025].

### 4.2. Desenvolvimento Iterativo (V1–V6)

A implementação seguiu uma abordagem incremental ao longo de seis versões, nas quais cada falha arquitetural diagnosticada motivou uma correção específica na versão seguinte. A Tabela 1 sintetiza essa trajetória, relacionando a principal mudança de cada versão ao seu efeito sobre o comportamento do sistema.

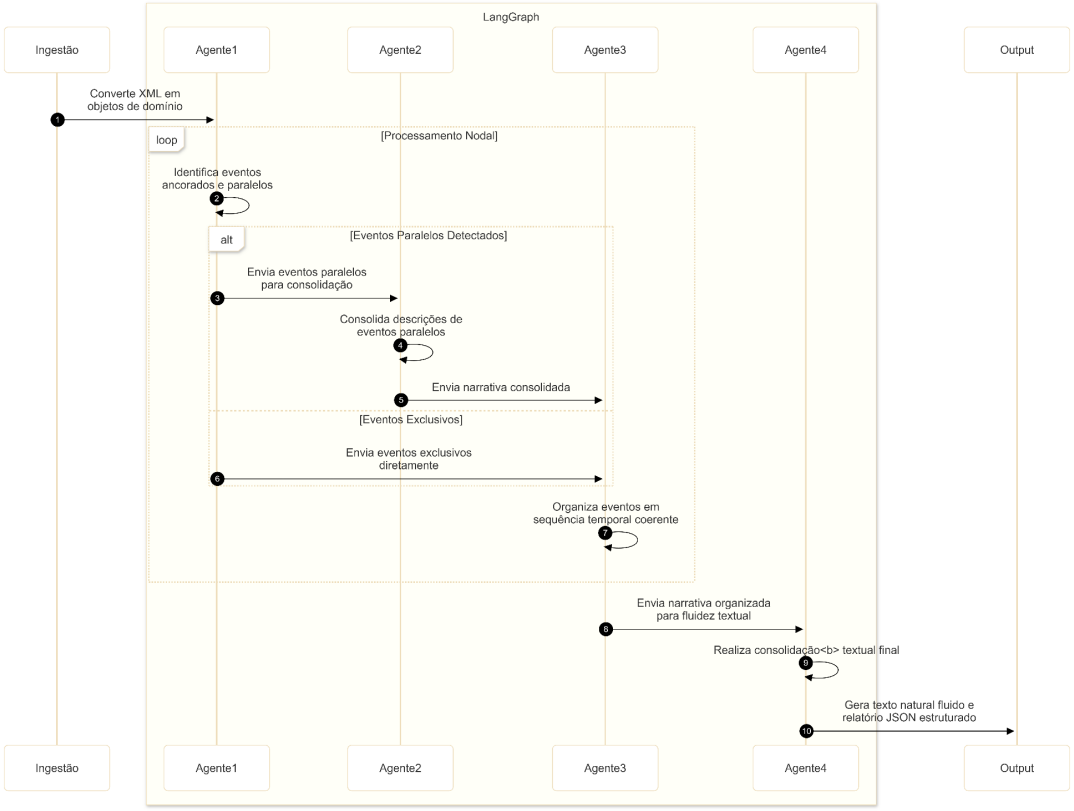


**Figura 2. Fluxo de Orquestração dos Agentes.**

**Tabela 1. Trajetória de desenvolvimento da pipeline (V1–V6).**

| Versão | Mudança arquitetural | Efeito observado |
|---|---|---|
| V1 (MVP) | Prova de conceito da consolidação em amostra limitada | Consolidação básica validada; eventos exclusivos não tratados |
| V2 | Gestão de memória e estado compartilhado | Estabilização do fluxo; ainda restrito a amostras |
| V3 | Execução completa centrada em Mateus | "Buraco Metodológico": descarte silencioso de eventos exclusivos das fontes secundárias |
| V4 | Estratégia de *Harvesting* para eventos órfãos | Melhora textual, mas falha de persistência e vazamento de sintaxe JSON na narrativa |
| V5 | Correção do JSON | Erro de marcação infla o *recall*: *timeline* com 69 eventos por duplicação |
| V6 | Deduplicação estrita via chaves compostas | 66 eventos únicos e 100% de cobertura do corpus |

A trajetória evidencia que os principais ganhos não decorreram de mudanças no modelo de linguagem, mantido fixo (GPT-4o) em todas as versões, mas de correções na arquitetura de recuperação e no controle de estado. O ponto de inflexão foi a V3, cujo diagnóstico do descarte silencioso de eventos exclusivos motivou a separação entre as etapas de recuperação e de ordenação, base da estratégia de *Full Coverage* consolidada

na V6. Esses ganhos decorreram de correções arquiteturais, e não de mudanças no modelo de linguagem, mantido fixo (GPT-4o) em todas as versões.

### 4.3. Protocolo de Avaliação

A avaliação adotou abordagem mista:

- **Testes quantitativos:** ROUGE-1, ROUGE-2 e ROUGE-L [Lin 2004] para sobreposição de $n$-gramas; BERTScore (F1) [Zhang et al. 2020b] para similaridade semântica via *embeddings* contextuais; e Kendall's Tau ($\tau$) [Kendall 1938] para coerência cronológica da ordenação dos eventos. A Pipeline foi comparada ao *golden sample* e a LLMs de estado da arte em regime *zero-shot* (prompt único).
- **Avaliação qualitativa:** Teste cego conduzido por especialista (teólogo), sem conhecimento sobre a origem dos textos avaliados, comparando o texto final da Pipeline V6 com a saída do ChatGPT-5.2 Pro (selecionado por apresentar os melhores resultados entre os LLMs de *baseline*).
  A avaliação foi conduzida com um instrumento padronizado de questões abertas, cobrindo ordem cronológica, completude, fidelidade factual e fluidez, projetado para ser replicável por outros avaliadores.

A execução completa da pipeline foi monitorada via LangFuse, totalizando ≈2h53min de processamento (26/01/2026, 19h54 às 22h47), consumo de ≈230.257 tokens e custo de USD 1,07, evidenciando a viabilidade computacional da abordagem.

Para garantir a reprodutibilidade dos resultados, o código-fonte da pipeline, o corpus em XML dos quatro Evangelhos, o *golden sample* utilizado na avaliação e os *system prompts* completos de cada agente (Ancorador, Consolidador e Fluência) foram disponibilizados em repositório público: `https://github.com/afcc2000/Four-Gospels-Agentic-MDS-NC.git`.

## 5. Resultados

### 5.1. Evolução Quantitativa da Pipeline

A Tabela 2 resume o desempenho das seis versões frente ao *golden sample*. A análise dos três grupos de métricas revela aspectos distintos do comportamento do sistema.

O ROUGE cresceu de forma consistente, o ROUGE-1 saltou de 0,53 na V1 para 0,7785 na V6, ganho atribuível diretamente à estratégia de *Full Coverage*: à medida que os eventos exclusivos deixaram de ser descartados, o texto final passou a conter uma parcela maior da informação presente na referência.

O BERTScore acompanhou essa trajetória, subindo de 0,9086 na V1 para 0,9379 na V6. A decomposição da métrica localiza o ganho: a precisão permaneceu praticamente constante em todas as versões, entre 0,931 e 0,937, enquanto o recall saltou de 0,8837 na V1 para 0,9432 na V6 , transição concentrada entre a V2 e a V3, a primeira execução sobre o corpus completo. A estabilidade da precisão indica que a capacidade de compreensão semântica do modelo base (GPT-4o) já era robusta desde o início e não constituía o gargalo do problema; o que mudou, e o que o recall mede, foi a arquitetura de recuperação.

O Kendall's Tau expõe o principal *trade-off* do projeto. O maior valor foi obtido na V3 (0,6696), justamente porque essa versão seguia estritamente a ordem do Evangelho

de Mateus e ignorava os eventos divergentes, coerência temporal alta, porém às custas da completude. A redução para 0,5571 na V6 é, portanto, um efeito esperado e assumido: ao anexar os eventos órfãos ao final da narrativa para assegurar cobertura integral, o sistema penaliza a linearidade cronológica em favor da integridade informacional.

**Tabela 2. Evolução das Métricas do Pipeline (V1–V6).**

| Versão | K. Tau | ROUGE-1 | ROUGE-2 | ROUGE-L | BERTScore |
|---|---|---|---|---|---|
| V1 (MVP) | 0,2360 | 0,5310 | 0,4357 | 0,3324 | 0,9086 |
| V2 | 0,5315 | 0,5379 | 0,4278 | 0,3388 | 0,9053 |
| V3 | 0,6696 | 0,6319 | 0,5694 | 0,4249 | 0,9369 |
| V4 | 0,5823 | 0,7633 | 0,7268 | 0,4674 | 0,9415 |
| V5 | 0,5529 | 0,7635 | 0,7358 | 0,4740 | 0,9438 |
| **V6** | **0,5571** | **0,7785** | **0,7496** | **0,4903** | **0,9379** |

## 5.2. Comparativo com LLMs em Regime *Zero-Shot*

A Tabela 3 compara a Pipeline V6 com modelos de estado da arte. A Pipeline atingiu o maior índice de fidelidade semântica (BERTScore 0,9379), superando todos os LLMs isolados. O salto sobre o modelo base (ROUGE-1 de 0,1921 no ChatGPT-4o puro para 0,7785 na Pipeline) comprova que a qualidade da harmonização depende fundamentalmente da engenharia de fluxo, não apenas da capacidade do modelo subjacente. A métrica de Kendall's Tau não foi aplicada aos LLMs de *baseline*, pois produzem texto narrativo livre sem metadados de ordenação de eventos rastreáveis.

**Tabela 3. Comparativo de Desempenho: Pipeline V6 vs. LLMs (*zero-shot*).**

| Modelo / Sistema | ROUGE-1 | ROUGE-2 | ROUGE-L | BERTScore |
|---|---|---|---|---|
| **Pipeline V6** | **0,7785** | **0,7496** | **0,4903** | **0,9379** |
| ChatGPT-4o | 0,1921 | 0,1386 | 0,1414 | 0,8794 |
| ChatGPT-5.2 Pro | 0,7482 | 0,4585 | 0,4765 | 0,9013 |
| Copilot GPT-5.2 | 0,6351 | 0,3156 | 0,3324 | 0,8699 |
| Gemini 3 Pro | 0,2818 | 0,2065 | 0,2120 | 0,8896 |

## 5.3. Avaliação Qualitativa

A avaliação cega, conduzida pelo especialista descrito na Seção 4, contrastou a saída da Pipeline V6 (Amostra A) com a do ChatGPT-5.2 Pro (Amostra B) em três eixos: fidelidade estilística, cobertura e harmonização entre fontes, e coerência cronológica.

**Fidelidade estilística.** A Amostra A foi descrita como mais formal e próxima ao registro do texto bíblico original, preservando o discurso direto (e.g., *"Jesus said..."*) e mantendo o relato em terceira pessoa sem interrompê-lo com referências aos autores. A Amostra B, embora mais fluida e acessível, recorreu a meta-comentários que atribuem o conteúdo aos evangelistas e quebram a imersão, como em *"Luke's memory preserves the content of that last surrender..."*. Segundo o especialista, esse tipo de moldura editorial não apenas prejudica o fluxo, como pode introduzir afirmações não sustentadas pelas fontes, aproximando-se de alucinação.

**Cobertura e harmonização.** O especialista destacou a capacidade da Pipeline de integrar relatos complexos. Na Entrada Triunfal, a Amostra A fundiu com êxito as versões

sinóticas com os detalhes próprios do Evangelho de João, ao passo que a Amostra B tratou o episódio de forma superficial. Identificou-se, porém, uma limitação relevante, denominada *viés de João*: em trechos de alta divergência entre as fontes, como a Ressurreição e os Discípulos de Emaús, a Pipeline tendeu a descartar os relatos sinóticos em favor da versão joanina. A narrativa das mulheres no sepulcro, presente em Mateus, Marcos e Lucas, foi suprimida em favor do relato de Maria Madalena em João. Esse comportamento é coerente com a lógica do Agente de Consolidação, que, diante de versões conflitantes, privilegia a fonte de maior extensão textual em vez de mesclar os detalhes complementares.

**Coerência cronológica.** O terceiro eixo confirmou, de forma independente, o diagnóstico associado ao Agente Organizador. O episódio dos gregos que desejavam ver Jesus (João 12) apareceu deslocado para o final da narrativa, após o julgamento de Pilatos; o avaliador interpretou corretamente o sintoma, observando que o sistema "não conseguiu encaixar em lugar nenhum e foi deixando para frente" — descrição precisa do tratamento dado aos eventos órfãos. A Expulsão dos Vendilhões, posicionada logo após a Entrada Triunfal, gerou descontinuidade, pois o sistema não reconheceu que a ida ao Templo encerra a Entrada Triunfal em Mateus. A Unção em Betânia, por sua vez, surgiu após a Entrada Triunfal, invertendo a relação de causa e efeito do texto original.

**Veredito.** A síntese do especialista reflete o *trade-off* central do projeto: a Amostra A é exegeticamente superior em detalhe e estilo, recuperando passagens exclusivas que o modelo comercial ignora, mas falha na macroestrutura, exigindo refinamento na ordenação temporal para que eventos sem âncora clara não sejam empurrados para o final nem fontes sinóticas preteridas pela densidade textual de João. Por se basear em um questionário unificado, o protocolo é replicável e pode ser estendido a um painel de especialistas em trabalhos futuros.

## 6. Limitações Metodológicas

A análise evidenciou dois gargalos principais: (1) **Dependência de segmentação prévia** (Agente 1), o sistema requer que os eventos estejam pré-delimitados em perícopes, limitando a autonomia em documentos não estruturados; (2) **Fragilidade do determinismo** (Agente 3), por ser um algoritmo Python, posiciona eventos com base no momento de descoberta pelo algoritmo e não pela lógica histórica, resultando nos "Eventos Órfãos" que impactam o Kendall's Tau.

O estudo foi conduzido em um corpus único e específico; os resultados devem ser lidos como exploratórios. Como trabalhos futuros, recomendamos:

(i) substituir o algoritmo determinístico do Agente 3 por um mecanismo de ordenação temporal mais robusto, seja um grafo multi-relacional inspirado no *Temporal Alignment Event Graph* (TAEG) [Finger et al. 2025], seja um LLM capaz de identificar marcadores temporais textuais para inferir a posição cronológica de eventos órfãos. Diferentemente do TAEG, que é extrativo e depende de uma cronologia canônica externa, a integração proposta manteria a fusão abstrativa do Agente 2, usando o grafo apenas como esqueleto estrutural para o posicionamento dos eventos órfãos. Os próprios autores do TAEG apontam a indução automática de cronologia como um problema em aberto na área;

(ii) realizar *benchmarking* com diferentes LLMs nos Agentes 1, 2 e 4;

(iii) investigar abordagens de segmentação automática de perícopes, por exemplo via modelos de detecção de fronteira de discurso ou classificadores supervisionados treinados em corpora anotados, de modo a eliminar a dependência atual de pré-segmentação manual em XML. Essa limitação também pode ser explorada pela introdução de um Agente 0, que usaria um LLM para preparar o corpus bruto no formato exigido pelo restante da *pipeline*, ampliando a autonomia do sistema para documentos não estruturados;
(iv) adaptar a *pipeline* para domínios como jurimetria, saúde, historiografia e *compliance*.

## 7. Considerações Finais

Esta pesquisa confirmou, de forma parcial e qualificada, que a ancoragem temporal e semântica de eventos combinada à consolidação abstrativa gera MDS mais coerente e completa do que abordagens não ancoradas.

A *pipeline* multiagêntica desenvolvida superou modelos de linguagem de estado da arte em regime *zero-shot* em fidelidade semântica (BERTScore 0,9379 vs. 0,9013 do melhor baseline), evidenciando que a qualidade da harmonização depende fundamentalmente da engenharia de fluxo e não apenas da capacidade do modelo subjacente. As avaliações qualitativas corroboraram a superioridade do texto em fidelidade estilística e riqueza exegética. As limitações concentraram-se na ordenação temporal global, sem comprometer a integridade factual ou a completude do material produzido.

Conclui-se que a orquestração de agentes especializados constitui um caminho eficaz para transformar LLMs em sistemas confiáveis de consolidação documental multidocumento, com potencial de impacto em historiografia digital, jornalismo computacional, análise documental e demais domínios que demandem fusão semântica com preservação da completude das fontes originais.

## Referências


Barzilay, R., McKeown, K. R., and Elhadad, M. (1999). Information fusion in the context of multi-document summarization. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 550–557, College Park, Maryland, EUA.

Bedi, H., Patil, S., Hingmire, S., and Palshikar, G. (2017). Event timeline generation from history textbooks. In *Proceedings of the 4th Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA 2017)*, pages 69–77, Taipei, Taiwan.

Biblica, Inc. (2011). The holy bible, new international version. Acesso em: 16 out. 2025.

Cunha, A. (2025). Semana da paixão unificada: PLN e ancoragem temporal na harmonização dos evangelhos. *Anais do IV Congresso Brasileiro de Humanismo Solidário na Ciência*.

Erkan, G. and Radev, D. R. (2004). LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479.

Finger, R. A., Cortes, E. G., Rigo, S. J., and de O. Ramos, G. (2025). Narrative consolidation: Formulating a new task for unifying multi-perspective accounts.

Kendall, M. G. (1938). A new measure of rank correlation. *Biometrika*, 30(1/2):81–93.

LangChain Team (2024). LangGraph documentation. Acesso em: 2025.

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out*, pages 74–81, Barcelona, Espanha.

Ma, C., Zhang, W. E., Guo, M., Wang, H., and Sheng, Q. Z. (2023). Multi-document summarization via deep learning techniques: A survey. *ACM Computing Surveys*, 55(5):1–37.

Mamidala, K. K. and Sanampudi, S. K. (2021). A novel framework for multi-document temporal summarization (MDTS). *Emerging Science Journal*, 5(2):184–190.

Petersen, W. L. (1994). *Tatian's Diatessaron: Its Creation, Dissemination, Significance, and History in Scholarship*. Brill, Leiden.

Russell, S. J. and Norvig, P. (2021). *Artificial Intelligence: A Modern Approach*. Pearson, Hoboken, 4 edition.

Santana, B., Campos, R., Amorim, E., Jorge, A., Silvano, P., and Nunes, S. (2023). A survey on narrative extraction from textual data. *Artificial Intelligence Review*, 56(8):8393–8435.

Song, J., Akhter, M. E., Atzil-Slonim, D., and Liakata, M. (2025). Temporal reasoning for timeline summarisation in social media. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 28085–28101, Viena, Áustria.

Wooldridge, M. (2009). *An Introduction to Multiagent Systems*. Wiley, Chichester, 2 edition.

Xiao, W., Beltagy, I., Carenini, G., and Cohan, A. (2022). PRIMERA: Pyramid-based masked sentence pre-training for multi-document summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4016–4036, Dublin, Irlanda.

Yasunaga, M., Zhang, R., Meelu, K., Pareek, A., Srinivasan, K., and Radev, D. (2017). Graph-based neural multi-document summarization. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 452–462, Vancouver, Canadá.

Yu, Y., Jatowt, A., Doucet, A., Sugiyama, K., and Yoshikawa, M. (2021). Multi-timeline summarization (MTLS): Improving timeline summarization by generating multiple summaries. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 377–387, Online.

Zhang, J., Zhao, Y., Saleh, M., and Liu, P. J. (2020a). PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning*, pages 11328–11339, Viena, Áustria.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2020b). BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*.