UniChat: Arquitetura e Avaliação de um Agente RAG de Baixo Custo para o Contexto Universitário

Ana Clara Boniatti Bordin, Gabriel Machado Lunardi, Eduardo K. Piveta, Leonardo Emmendorfer

[1]Centro de Tecnologia – Universidade Federal de Santa Maria (UFSM)

{acbordin, piveta}@inf.ufsm.br

gabriel.lunardi@ufsm.br, leonardoemmendorfer@gmail.com

Abstract. *This paper introduces UniChat, a low-cost chatbot based on RAG to optimize access to university information. The solution utilizes free and open-source technologies like n8n, PostgreSQL, and Google Gemini. Its main contributions include the low-cost architecture , empirical analysis of the correlation between source document format and response accuracy , and validation of the user experience with Think Aloud and AttrakDiff Mini. Tests with real users demonstrated a 76.7% success rate and a positive perception of functionality and intuitiveness. UniChat has the potential to enhance university communication and reduce administrative overload.*

Resumo. *Este artigo apresenta o UniChat, chatbot de baixo custo baseado em RAG para otimizar o acesso à informação universitária. A solução utiliza tecnologias gratuitas e de código aberto como n8n, PostgreSQL e Google Gemini. As principais contribuições incluem a arquitetura de baixo custo, análise empírica da correlação entre o formato dos documentos-fonte e a acurácia das respostas, e validação da experiência do usuário com Think Aloud e AttrakDiff Mini. Testes com usuários reais demonstraram uma taxa de sucesso de 76,7% e percepção positiva de funcionalidade e intuitividade. O UniChat tem potencial para aprimorar a comunicação universitária e reduzir a sobrecarga administrativa.*

1. Introdução

No atual cenário digital, a dispersão e a imprecisão de informações em diversas plataformas dificulta o acesso rápido e centralizado a dados fidedignos e atualizados [Moreira et al. 2023]. Essa fragmentação e a frequente ausência de informações pontuais em canais ambientes universitários resultam na sobrecarga de setores administrativos e na dependência de fontes informais, comprometendo a eficiência da comunicação entre a instituição e a comunidade acadêmica [Zuboff 2019]. Diante dessa necessidade premente de acesso centralizado e ágil, este artigo apresenta o UniChat, um agente conversacional baseado em RAG (*Retrieval-augmented generation*) para otimizar o acesso à informação institucional universitária. Com isso, o objetivo é desenvolver uma solução de baixo custo, replicável e funcional, que promova maior agilidade e acessibilidade na comunicação.

As principais contribuições do UniChat incluem: (i) uma arquitetura de sistema de baixo custo baseada em tecnologias gratuitas e de código aberto; (ii) uma análise empírica que correlaciona o formato dos documentos-fonte com a acurácia das respostas; e (iii) a validação da experiência do usuário por meio de testes com usuários reais, utilizando métricas reconhecidas como Think Aloud e AttrakDiff Mini.

2. Trabalhos relacionados

Soluções baseadas em RAG são uma tendência como forma de gerar respostas contextuais e relevantes na academia e na indústria. A eficácia desse método foi avaliada na resolução de questões do Exame Nacional do Ensino Médio (ENEM), demonstrando aumento significativo de acurácia em tarefas de contextualização [Taschetto et al. 2024]. Ainda no contexto educacional, o Cosmobot foi desenvolvido como um chatbot para auxiliar estudantes no aprendizado de algoritmos por meio da resolução de questões de provas anteriores, oferecendo suporte pedagógico prático [Estrela et al. 2024]. Outros estudos aplicaram RAG em documentos jurídicos [Aquino et al. 2024] e desenvolveram metodologias de feedback implícito para monitorar a qualidade de aplicações corporativas [Albuquerque et al. 2024].

Chatbots também têm sido explorados em contextos universitários com abordagens diversas. O trabalho de [Azzolin 2022] introduziu o GURIBO, um chatterbot desenvolvido para auxiliar a secretaria do campus da Unipampa, focando principalmente em estágios. Já [Silva and Nascimento 2023] apresentaram o Polímata, um chatbot de código aberto voltado ao curso de Ciência da Computação da UNIFAP, com foco em perguntas frequentes e participação colaborativa dos alunos para expansão da base de conhecimento.

Diferentemente desses trabalhos, este trabalho se propõe a ser uma solução de baixo custo que une RAG com uma infraestrutura leve e acessível, aproveitando ferramentas como o n8n, PostgreSQL e o modelo Gemini da Google em plano gratuito. O foco do UniChat está na fidedignidade das respostas, acessibilidade dos dados e usabilidade da interface. Além disso, ele é um dos poucos que realiza testes com usuários reais, utilizando métricas UX reconhecidas, como Think Aloud e AttrakDiff, o que o posiciona como um experimento prático e validado em experiências reais de uso.

3. O Agente Conversacional UniChat

O UniChat tem como premissa ser uma ferramenta que utiliza serviços e planos gratuitos. O principal orquestrador é a plataforma N8N, que por ser *open source* oferece uma versão *self-hosted*. No contexto do Unichat, a plataforma é hospedada em uma máquina virtual do plano *Always Free* da Oracle Cloud Infrastructure, que garante gratuidade vitalícia. Além disso, o banco de dados PostgreSQL e a plataforma de contêineres Docker são ambos gratuitos e de código aberto. Por fim, o modelo de linguagem Google Gemini é consumido por meio do seu plano gratuito, assegurando que todas as etapas do protótipo possam ser executadas sem despesas.

A arquitetura do UniChat é estruturada em duas etapas: i. coleta automatizada de documentos institucionais armazenados em uma pasta no Google Drive, vetorização dos documentos utilizando *embeddings* de 768 dimensões, armazenamento dos vetores em banco PostgreSQL (com extensão PGVector) e ii. geração de respostas ao usuário final por meio do modelo Gemini 2.0-flash, que recupera trechos contextualmente relevantes diretamente dessa base de dados com memória vetorial.

Na plataforma N8N, estão implementados dois fluxos responsáveis pelo funcionamento do UniChat. O N8N é uma plataforma de automação focada em integrar serviços por meio de uma interface *no-code*, estruturada em nós que se conectam sequencialmente. Cada nó representa uma tarefa, como consumir uma API, processar dados ou interagir com um banco de dados; essa estrutura possibilita a construção de fluxos escaláveis e dinâmicos, tornando a ferramenta ideal para a construção do projeto.

3.1. O Fluxo de Armazenamento de Documentos

O fluxo de armazenamento de documentos demonstrado na Figura 1 tem como função automatizar a ingestão, o processamento e a indexação semântica de arquivos em diferentes formatos. Os autores [Lunardi et al. 2018] já destacavam que um dos principais impasses no desenvolvimento de agentes conversacionais é justamente a construção automática de suas bases de conhecimento. O UniChat contorna essa limitação ao converter os arquivos institucionais em vetores semânticos armazenados em banco vetorial, permitindo atualização contínua.

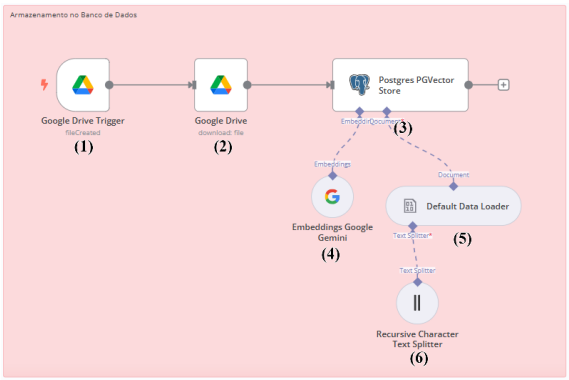

Figura 1. Captura de tela do fluxo de armazenamento na plataforma N8N

O armazenamento de documentos no UniChat começa com um nó gatilho que monitora a pasta “Chatbot Universidade” no Google Drive, como mostra o item 1 da Figura 1, disparando o fluxo sempre que um novo arquivo é adicionado. Após o download do arquivo, indicado no item 2, o conteúdo é convertido em texto pelo *Default Data Loader* (item 5), que adapta automaticamente o processamento dependendo do formato do documento. O texto é então particionado, isto é, tokenizado, em blocos menores no *Recursive Character Text Splitter* (item 6), com preservação de contexto através de *overlaps*, e vetorizado pelo modelo *text-embedding-004* da Google (item 4), gerando embeddings de 768 dimensões. Por fim, os vetores, textos e metadados são armazenados em uma tabela no banco PostgreSQL com extensão PGVector (item 3), formando uma base vetorial pronta para consultas e integrando automaticamente novos documentos à memória do chatbot.

3.2. O Fluxo de Geração de Respostas

A Figura 2 apresenta o fluxo responsável por converter a mensagem do usuário em uma resposta informativa baseada no paradigma RAG [Lewis et al. 2020].

O fluxo de geração de respostas do UniChat tem início no nó *When chat message received*, conforme mostra o item 1 da Figura 2, responsável por disparar o processo assim que uma nova mensagem é recebida. Em seguida, o *AI Agent*, representado no item 2, conecta-se ao modelo de linguagem, à memória de curto prazo e a um subfluxo de ferramentas, indicados respectivamente nos itens 3, 4 e 5. Esses elementos são guiados por um *system prompt* que orienta a priorização da base vetorial, o uso de metadados atualizados e a limitação do conteúdo às informações da universidade. A etapa posterior é coordenada pelo modelo Google Gemini, destacado no item 3 da figura, que atua como

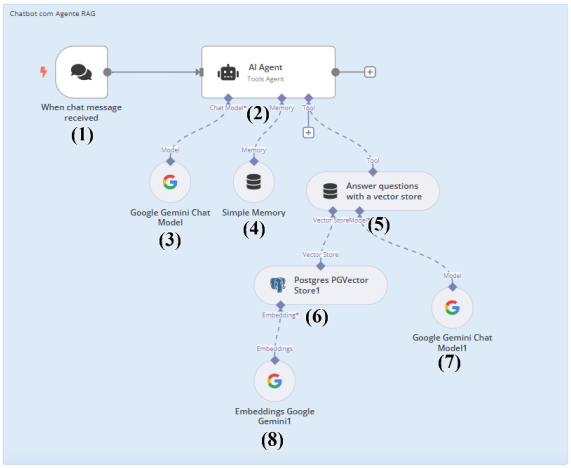

Figura 2. Captura de tela do fluxo de geração de respostas na plataforma N8N

orquestrador, analisando a intenção da consulta e decidindo entre utilizar o contexto recente registrado na *Simple Memory*, como indicado no item 4, ou acionar a ferramenta *Answer questions with a vector store*, conforme representado no item 5. A consulta é então convertida em um *embedding* de 768 dimensões no nó *Embeddings Google Gemini*, conforme ilustrado no item 8, e enviada ao banco *PGVector Store*, mostrado no item 6, que retorna os fragmentos mais relevantes da tabela *n8n_vectors*. Por fim, esses trechos, junto à pergunta original, são encaminhados ao modelo generativo Gemini, localizado no item 7 da Figura 2, que elabora a resposta final com base nas evidências recuperadas, assegurando precisão e atualidade às informações fornecidas sobre a instituição.

4. Avaliação e Resultados

Existem limitações nas métricas automatizadas tradicionais de Pergunta e Resposta pois elas não abrangem as nuances de contexto, frequentemente falhando em capturar a compreensão semântica profunda, a fluidez textual e a veracidade das informações [Kamalloo et al. 2023]. *Frameworks* de avaliação RAG costumam focar na etapa de *retrieval*, além de demonstrarem ter correlação muito mais baixa do que a avaliação humana [Oro et al. 2024].

Logo, para avaliar a relevância e a acurácia das respostas do Unichat, foram escolhidos dois métodos de avaliação da experiência do usuário: o Think Aloud [Barbosa et al. 2022a] e o AttrakDiff Mini [Vieira et al. 2023, Barbosa et al. 2022b]. O AttrakDiff Mini é um questionário de dez pares de adjetivos separados em três categorias: “PQ” sobre utilidade e clareza, “HQ” sobre prazer e identidade e “ATT” sobre a atratividade geral da interface. Já o Think Aloud é um método no qual o usuário verbaliza seus pensamentos enquanto utiliza um sistema para tornar possível que o condutor compreenda, analise e metrifique suas percepções.

Os testes foram conduzidos com dez participantes, dos quais oito eram estudantes de diferentes áreas da universidade e dois eram externos: uma profissional da advocacia e uma estudante de odontologia de outra instituição. As sessões ocorreram em chamadas virtuais individuais com o condutor da pesquisa. O método Think Aloud foi sempre aplicado primeiro, seguido pelo questionário do AttrakDiff Mini, pois o Think Aloud

permite capturar percepções espontâneas durante o uso guiado, enquanto o AttrakDiff quantifica essas impressões logo após a experiência.

Durante o Think Aloud, os participantes foram instruídos a realizarem seis perguntas ao UniChat, abrangendo diferentes escopos: três perguntas baseadas em informações que o *chatbot* possui ("Descubra quando iniciam as aulas do segundo semestre"), uma contendo erro ortográfico propositalmente e duas perguntas sobre temas que o Unichat não tem informação. É importante destacar que os participantes não foram informados sobre quais dados alimentavam o chatbot, a fim de evitar qualquer viés no comportamento. Ao final da etapa interativa, cada participante também respondeu a uma pergunta de opinião sobre a interface (em escala de 1 a 5) e a uma questão sobre o que mudaria no geral, se pudesse. Além disso, o condutor documentou quais respostas foram respondidas como esperado e quais falharam.

Das 60 tarefas realizadas pelos dez participantes, o UniChat apresentou uma taxa de sucesso de 76,7%. A média das notas atribuídas à interface na escala de 1 a 5 foi de 4,3, com sugestões de melhoria visual, como o ajuste no tom de azul e o alinhamento de elementos. A seção de análise qualitativa indicou que o sistema foi percebido positivamente e descrito com frequência como direto e funcional.

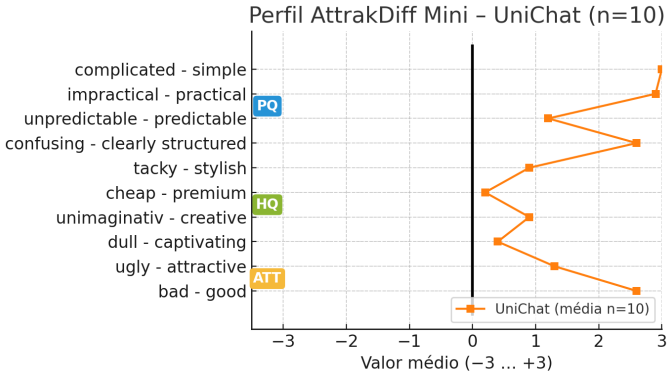

Figura 3. Resultado do método AttrakDiff

No AttrakDiff Mini, apresentado na Figura 3, todas as dimensões ficaram acima do ponto neutro (zero). A dimensão "PQ" (Qualidade Pragmática) obteve as maiores pontuações, indicando que o UniChat foi percebido como funcional e intuitivo. A "ATT" (Atratividade Global) também foi avaliada com uma boa impressão geral sobre o sistema, o que corrobora com os comentários registrados durante o Think Aloud. Já a "HQ" (Qualidade Hedônica) apresentou as menores médias entre os três critérios, sugerindo que os aspectos estéticos e de identidade do sistema foram percebidos como apenas moderadamente positivos, apontando uma oportunidade de aprimoramento visual e de engajamento emocional.

Após a obtenção destes dados quantitativos sobre as perguntas que mais receberam respostas erradas na etapa do Think Aloud, foi possível analisá-las isoladamente a fim de descobrir quais critérios levaram à falha na resposta. Embora o Gemini realize corretamente a etapa de *retrieval* identificando as palavras-chave da pergunta e acessando o documento relevante por meio do nó de banco de dados, ele nem sempre compreende corretamente o contexto extraído. Isso ocorre, em parte, porque os documentos originalmente em

formato .DOCX passam pela conversão para .TXT, que gera diversas quebras de linha e fragmentações que dificultam a interpretação textual pela LLM, mesmo ela possuindo a informação.

Um exemplo recorrente foi a pergunta "Qual o e-mail e telefone da secretaria?", que o modelo interpretou corretamente em 60% dos casos. Embora ele tenha recuperado as informações corretas, não compreendeu que os termos "e-mail" e "telefone" eram sinônimos contextuais de "contato", como está rotulado nos documentos. Esse tipo de abstração semântica depende de uma interpretação contextual mais robusta, que o Gemini nem sempre alcançou, especialmente em consultas que envolviam siglas como "RI" (o curso de Relações Internacionais) ou menções a "EAD" (ensino a distância).

Por outro lado, perguntas mais literais e com termos diretamente alinhados ao conteúdo do documento, como "Quando iniciam as aulas do segundo semestre?", foram respondidas corretamente em 100% dos casos. Isso sugere que, embora a arquitetura de recuperação esteja bem arquitetada para recuperação correta das informações necessárias, o desempenho geral do UniChat poderia se aproximar de uma acurácia total com a adoção de uma LLM mais avançada em compreensão semântica e de interpretação contextual.

5. Considerações Finais

O UniChat demonstrou com sucesso a viabilidade de promover acessibilidade e autonomia no acesso à informação institucional universitária, mesmo em pequena escala e mantendo a fidelidade aos documentos oficiais. A avaliação empírica com usuários confirmou a percepção positiva do sistema como funcional e intuitivo. Em particular, observou-se que o formato dos documentos-fonte influencia diretamente a acurácia das respostas do chatbot, com perguntas literais respondidas com assertividade, enquanto a interpretação de sinônimos contextuais e siglas representou um desafio para o modelo atual.

Para trabalhos futuros, serão implementadas melhorias na interface e testes de alimentação da base de dados com arquivos no formato .TXT, a fim de comparar seu desempenho com os atuais .DOCX. A expectativa é que o uso de formatos nativamente textuais e bem estruturados contribua para uma maior fidelidade nas respostas. Os resultados obtidos indicam que o UniChat tem potencial para atuar como um aliado na comunicação universitária, com capacidade de reduzir a sobrecarga dos setores administrativos e otimizar a acessibilidade a informações institucionais.

Agradecimentos

Os autores agradecem ao CNPq pelo projetos universais 405973/2021-7 e 402086/2023-6, assim como à FAPERGS pelo projeto ARD/ARC – processo 24/2551-0000645-1.

Referências

Albuquerque, R. et al. (2024). Avaliação de aplicações de geração aumentada de recuperação por meio de feedback implícito. In *Anais Estendidos do XXXIX Simpósio Brasileiro de Bancos de Dados (SBBD)*, Porto Alegre, RS.

Aquino, J. et al. (2024). Extracting information from brazilian legal documents with retrieval augmented generation. In *Anais Estendidos do XXXIX Simpósio Brasileiro de Bancos de Dados (SBBD)*, Porto Alegre, RS.

Azzolin, J. a. V. (2022). Guribo: Chatterbot para auxílio à secretaria acadêmica do campus alegrete. Trabalho de Conclusão de Curso, Universidade Federal do Pampa.

Barbosa, M., Nakamura, W. T., Valle, P., Guerino, G. C., Finger, A. F., Lunardi, G. M., and Silva, W. (2022a). Ux of chatbots: An exploratory study on acceptance of user experience evaluation methods. In *Proceedings of the 24th International Conference on Enterprise Information Systems (ICEIS) – Volume 2*, pages 355–363. SciTePress.

Barbosa, M., Valle, P., Nakamura, W., Guerino, G., Finger, A., Lunardi, G., and Silva, W. (2022b). Um estudo exploratório sobre métodos de avaliação de user experience em chatbots. In *Anais da VI Escola Regional de Engenharia de Software*, Porto Alegre, RS, Brasil. SBC.

Estrela, I. R. B. et al. (2024). Ferramenta cosmobot: Um chatbot de apoio a alunos em avaliações de algoritmos. In *Anais do XXXV Simpósio Brasileiro de Informática na Educação (SBIE)*, pages 380–391, Rio de Janeiro, RJ.

Kamalloo, E., Dziri, N., Clarke, C. L. A., and Rafiei, D. (2023). Evaluating open-domain question answering in the era of large language models.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., and Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS)*, pages 9459–9474.

Lunardi, G. M., Machado, G. M., Al Machot, F., Maran, V., Machado, A., Mayr, H. C., Shekhovtsov, V. A., and de Oliveira, J. P. M. (2018). Probabilistic ontology reasoning in ambient assistance: predicting human actions. In *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*, pages 593–600. IEEE.

Moreira, L. S., Lunardi, G. M., de Oliveira Ribeiro, M., Silva, W., and Basso, F. P. (2023). A study of algorithm-based detection of fake news in brazilian election: Is bert the best. *IEEE Latin America Transactions*, 21(8):897–903.

Oro, E., Granata, F. M., Lanza, A., Bachir, A., Grandis, L. D., and Ruffolo, M. (2024). Evaluating retrieval-augmented generation for question answering with large language models. In *Proceedings of Ital-IA 2024 – 4th National Conference on Artificial Intelligence, Thematic Workshops*, volume 3762, pages 12–17, Naples, Italy.

Silva, L. and Nascimento, T. P. d. (2023). Chatbot polímata: Assistente aberto para informações do curso. Projeto da Universidade Federal do Amapá.

Taschetto, M. et al. (2024). Using retrieval-augmented generation to improve performance of large language models on the brazilian university admission exam. In *Anais do XXXIX Simpósio Brasileiro de Bancos de Dados (SBBD)*, Porto Alegre, RS.

Vieira, D. et al. (2023). Design of a smart garment for fencing: measuring attractiveness using the attrakdiff mini method. *Human-Intelligent Systems Integration*, 5(1–2):1–9.

Zuboff, S. (2019). *A era do capitalismo de vigilância: a luta por um futuro humano na nova fronteira do poder*. Intrínseca, Rio de Janeiro.