Ferramenta Inteligente para Gestão de E-mails Corporativos utilizando LLMs

Matheus Lorenzato Braga
Instituto Federal Catarinense
Campus Sombrio
Sombrio, SC / Brasil
matheus.braga@ifc.edu.br

Pedro Luiz Pompeu da Silva
Universidade do Estado de Santa Catarina
Campus Florianópolis
Florianópolis, SC / Brasil
pedrolpompeu@gmail.com

ABSTRACT

This paper presents the development and functional analysis of an application designed to optimize corporate communication through automatic email classification and response generation using the Google Gemini API. The study addresses the challenge of increasing message volumes in organizations, which negatively impacts productivity and response times. The adopted methodology was experimental, integrating a Python backend architecture with strategies to mitigate rate limitations. Tests were conducted using a dataset of 1000 messages to evaluate response times and classification accuracy between productive and unproductive emails. The results demonstrate that the solution provides efficiency gains in sorting and response standardization, validating the use of advanced language models in corporate environments, despite computational cost limitations at large scales.

KEYWORDS

Automação de E-mail, Google Gemini, Inteligência Artificial, LLM, Processamento de Linguagem Natural.

1 Introdução

A crescente utilização do correio eletrônico nas organizações cria oportunidades e desafios relacionados à gestão da comunicação corporativa. A interconectividade entre funcionários, clientes, fornecedores e outros públicos de interesse favorece a troca de informações de maneira rápida, barata e documental. Para as organizações, o relacionamento contínuo com o cliente é um caminho para a captação de informações valiosas sobre quem é o cliente, suas expectativas, suas necessidades atuais e futuras [4].

Entretanto, o volume crescente de mensagens eletrônicas nas organizações representa um desafio recorrente na gestão da comunicação. A filtragem manual de e-mails, além de consumir tempo significativo, tende a provocar atrasos em processos decisórios e comprometer a produtividade das equipes. Nesse contexto, soluções automatizadas tornam-se uma alternativa promissora para aprimorar o fluxo de trabalho.

Com o avanço das tecnologias de análise de linguagem e modelos de predição contextual, emergem sistemas capazes de compreender o conteúdo semântico dos textos [3]. Diferentemente das abordagens tradicionais de filtragem, baseadas em regras fixas ou palavras-chave, os LLMs são capazes de capturar nuances contextuais e adaptar suas respostas ao tom e ao propósito de cada mensagem [2]. A aplicação proposta foi desenvolvida considerando essa tendência, integrando um modelo de linguagem avançado para classificar e-mails segundo critérios de produtividade e sugerir respostas contextualizadas.

Neste trabalho, apresenta-se uma ferramenta que integra estratégias de Processamento de Linguagem Natural (PLN) e os Grandes Modelos de Linguagem (LLMs) [1], para organizar semanticamente as informações e apoiar o processo de tomada de decisão. Dentre os modelos disponíveis atualmente, destacam-se aqueles oferecidos por grandes provedores de tecnologia, como o Google. A escolha pelo uso da API (*Application Programming Interface*) Google Gemini [5] fundamenta-se na capacidade do modelo em interpretar texto e fornecer respostas com padrões comunicativos. Além disso, os autores em [3] apresentaram resultados importantes, com descrições simples e em cenários avançados, os modelos da família Gemini demonstraram um desempenho superior.

A integração entre o modelo e a aplicação web permitiu criar uma solução acessível, escalável e segura para ambientes corporativos. Este trabalho analisa o ciclo de desenvolvimento do sistema, desde o planejamento da arquitetura de software até os testes de desempenho e análise dos resultados. O estudo também propõe reflexões sobre o impacto da automação de respostas na dinâmica de comunicação empresarial.

2 Metodologia

A metodologia adotada teve caráter experimental e aplicado, centrada na prototipagem funcional da aplicação e na avaliação empírica do seu desempenho. O projeto foi desenvolvido em Python para o backend, com HTML5, CSS3 e JavaScript na camada de interface. A interface foi estruturada de forma responsiva, priorizando clareza e facilidade de navegação.

A integração com a API Google Gemini foi feita por meio da biblioteca *google-generativeai*[1], tendo o modelo Gemini 2.5 Flash como responsável pela comunicação direta com o mecanismo de interpretação textual. Foram realizados testes com diferentes parâmetros de temperatura e emissão de respostas, buscando equilibrar consistência semântica e naturalidade na linguagem gerada. Para lidar com eventuais interrupções causadas por limites de taxa da API, incorporou-se uma lógica de retentativa com *Exponential Backoff*.

[1] Disponível em: https://pypi.org/project/google-generativeai/

Os experimentos foram conduzidos no *Google Colaboratory* (Colab), utilizando o ambiente de execução padrão disponibilizado gratuitamente pela plataforma, com processador virtual compartilhado e sem acelerador de hardware dedicado (GPU/TPU), com Python como linguagem principal para a implementação do protótipo e a conexão com a API. O modelo foi configurado com temperatura de 0,5, buscando um ponto de equilíbrio entre coerência nas respostas e capacidade interpretativa. O processo de avaliação teve caráter exploratório, com foco em verificar a viabilidade da abordagem na categorização automática de mensagens entre produtivas e improdutivas.

Para essa classificação, foi construído um conjunto de dados com 1000 mensagens de e-mails. As mensagens não foram rotuladas previamente nas categorias. Para as categorias, considerou-se "Produtivo" todo e-mail com demandas operacionais, solicitações técnicas ou comerciais, enquanto "Improdutivo" abrangeu comunicações informais, redundantes ou sem impacto operacional direto. Vale destacar que o conjunto não foi balanceado entre as categorias, o que pode introduzir viés na distribuição dos resultados e será endereçado em versões futuras com a construção de um corpus anotado manualmente. O sistema utilizou esses critérios para calibrar o comportamento da API e verificar a coerência das respostas geradas.

A aplicação foi ainda adaptada para aceitar arquivos nos formatos .txt e .pdf, ampliando o escopo de conteúdos analisáveis. Implementou-se também um controle de erros para tratar falhas de leitura, requisições mal formadas e problemas de timeout. A arquitetura geral da implementação pode ser observada na Figura 1.

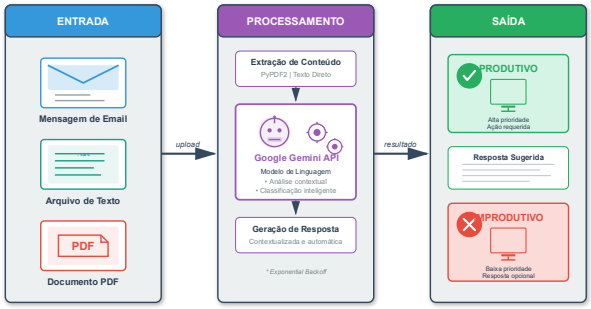

Figura 1: Arquitetura da ferramenta

Para a avaliação de desempenho, foram definidos como indicadores o tempo médio de resposta por requisição, a mediana e o desvio padrão dos tempos registrados, buscando caracterizar tanto a eficiência central quanto a variabilidade do sistema sob diferentes cargas. A taxa de acerto na categorização não foi avaliada nesta etapa, uma vez que o conjunto de dados não foi rotulado previamente. Métricas como acurácia, precisão e F1-score serão incorporadas em trabalhos futuros, com a construção de um *dataset* rotulado manualmente. Atualmente, a ferramenta permite a análise de um e-mail por vez, seja por meio de inserção manual do texto ou upload de arquivos individuais.

3 Resultados e Discussão

Este trabalho apresenta resultados preliminares de uma ferramenta ainda em desenvolvimento, com avaliação de caráter exploratório, centrada na análise da coerência semântica das classificações. Por se tratar de uma prova de conceito, não foi realizada validação estatística a partir da rotulagem dos dados, o que configura uma limitação metodológica reconhecida. A proporção exata entre e-mails produtivos e improdutivos não foi registrada durante a coleta; a construção de um *dataset* rotulado manualmente e o cálculo de métricas como acurácia, precisão e F1-score estão previstos como etapas futuras do projeto.

Vale destacar que os resultados foram obtidos sem técnicas clássicas de PLN, como *stemming* ou remoção de *stop words*, o que evidencia a capacidade contextual da API Google Gemini. A implementação do *Exponential Backoff* também se mostrou eficaz, reduzindo falhas por limitação de taxa e mantendo a estabilidade operacional.

Para avaliação do desempenho, foram definidos cenários com 10, 20, 25 e 50 requisições, processadas automaticamente via *script*. Os indicadores registrados para cada grupo incluem média, desvio padrão, mediana, tempo mínimo e tempo máximo de resposta, apresentados na Tabela 1.

Tabela 1: Média de tempo por requisições

Requisições	Média	Mediana	Desvio padrão
10	1,96s	1.91s	0.43s
20	1,97s	1.71s	0.62s
25	2,18s	1.81s	0.93s
50	2,49s	1.94s	1.39s

A Figura 2 mostra como os tempos de resposta se distribuíram nos quatro cenários testados. Nos grupos de 10 e 20 requisições, o comportamento foi mais regular, com medianas de 1,91s e 1,71s e desvios padrão de 0,43s e 0,62s, respectivamente. Com o aumento do volume de requisições, tanto a média quanto a variação dos tempos cresceram: o grupo de 25 requisições teve mediana de 1,81s e desvio padrão de 0,93s; já no grupo de 50, a mediana foi de 1,94s e o desvio chegou a 1,39s. O tempo máximo registrado nesse último grupo foi de 7,88s, o que aponta para a presença de outliers em situações de carga mais alta, provavelmente causados por restrições de taxa da API ou instabilidades no ambiente de execução compartilhado do Google Colab.

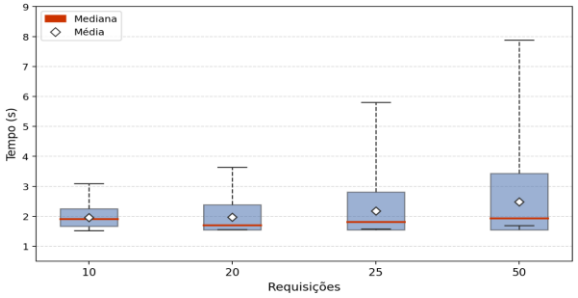

Figura 2: Distribuição dos tempos de resposta das requisições

Além da análise de desempenho, foi desenvolvida uma interface para interação com a ferramenta, com suporte a modos claro e escuro e funcionalidades como cópia rápida da resposta sugerida, sem que tenham sido conduzido um teste formal nesta etapa. A Figura 3 ilustra um exemplo de e-mail inserido no analisador.

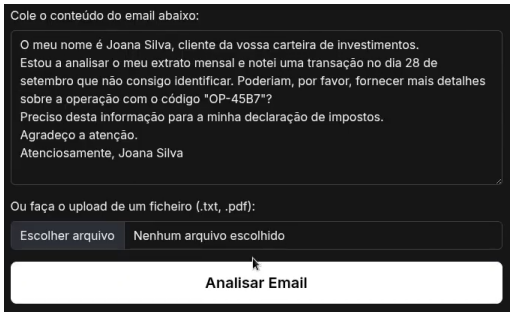

Figura 3: Interface para envio do e-mail ao analisador

Do ponto de vista funcional, a ferramenta pode contribuir para maior agilidade no atendimento interno, reduzindo o tempo de triagem. A classificação automática permite priorizar comunicações de valor comercial ou técnico, enquanto as respostas sugeridas ajudam a manter padrões consistentes de comunicação. As Figuras 4 e 5 mostram o processo de classificação, com a identificação das categorias Produtivo e Improdutivo, respectivamente.

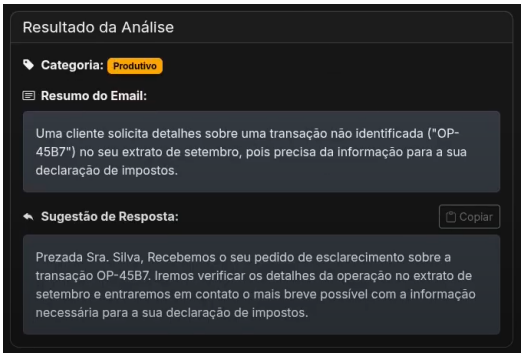

Figura 4: E-mail classificado como "Produtivo"

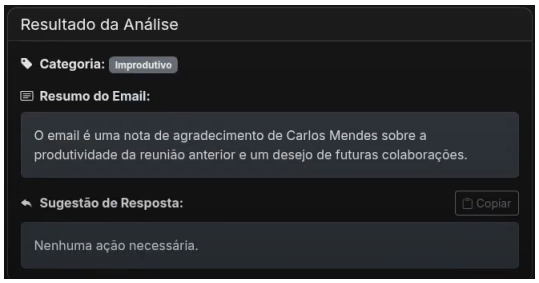

Figura 5: E-mail classificado como "Improdutivo"

A interface foi projetada para facilitar a interação do usuário, permitindo o envio de e-mails por meio de texto ou upload de arquivos. Na versão atual, o usuário pode submeter um e-mail por vez, seja colando o texto diretamente na interface ou realizando o upload de um arquivo individual nos formatos .pdf e .txt. Os testes realizados demonstraram tempos médios de resposta próximos de dois segundos.

Como limitação operacional, observou-se o custo computacional do uso contínuo da API em cenários de alta densidade de requisições. Para aplicações em maior escala, o uso de cache semântico para consultas repetitivas aparece como alternativa relevante. A adaptação das respostas a diferentes estilos de comunicação organizacional também se coloca como frente de desenvolvimento futuro.

Em síntese, os resultados reforçam que os LLMs são capazes de substituir etapas de pré-processamento textual sem perda expressiva de precisão, desde que operem dentro de uma arquitetura resiliente e monitorada - o que se alinha à tendência mais ampla de automação inteligente em processos corporativos.

4 Conclusões

O projeto proposto apresentou resultados preliminares de uma ferramenta em desenvolvimento. A ferramenta demonstrou que é possível otimizar a gestão de e-mails corporativos por meio da integração entre frameworks web tradicionais e tecnologias de análise linguística de última geração. O uso da API Google Gemini permitiu desenvolver uma ferramenta precisa, robusta e com capacidade de adaptação ao contexto das mensagens recebidas.

Os ganhos operacionais incluem redução do tempo de triagem, padronização das respostas e maior eficiência na priorização de tarefas. Embora o custo computacional continue sendo um fator a considerar, os resultados indicam que a solução é viável para ambientes corporativos de médio e grande porte.

Como trabalhos futuros, sugere-se explorar estratégias de personalização de respostas por perfil de remetente, bem como ampliar o escopo do sistema para suporte multilíngue e integração com plataformas de CRM, além de análises estatísticas para garantir a confiabilidade.

REFERÊNCIAS

[1] Bharathi Mohan, G., Prasanna Kumar, R., Vishal Krishh, P., Keerthinathan, A., Lavanya, G., Meghana, M. K. U., ... & Doss, S. (2024). An analysis of large language models: their impact and potential applications. Knowledge and Information Systems, 66(9), 5047-5070.

[2] Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., ... & Xie, X. (2024). A survey on evaluation of large language models. ACM transactions on intelligent systems and technology, 15(3), 1-45.

[3] Cruz, A., Rettore, P. H., & Santos, B. P. (2025, November). Análise do uso de Modelos de Linguagem de Grande Escala na Geração de Códigos para Automação Residencial. In Brazilian Symposium on Multimedia and the Web (WebMedia) (pp. 48-56). SBC.

[4] de Carvalho, D. T., & Vicari, F. M. (2001). Correio eletrônico e orientação para o mercado. Revista de Administra&ccdeil; ão da Universidade de São Paulo, 36(4).

[5] Dias, A. B. (2011). A gestão eletrónica documental como melhoria do fluxo de informação: Um estudo de caso.

[6] Google DeepMind. 2025. Gemini. https://deepmind.google/technologies/gemini/. Acessado em: jul. 2025.