

Projeto SARA: Sistema Automatizado de Resposta a Recursos dos Pedidos de Acesso à Informação

Douglas Rolins de Santana, Lívia Mancine Coelho de Campos,
Kairo Antônio Lopes da Silva, Danilo Silva Ramos,
Valdemar Vicente Graciano Neto, Leonardo Andrade Ribeiro

¹Instituto de Informática – Universidade Federal de Goiás (UFG) – Goiânia, GO – Brasil
{douglasrolins,kairoantonio,liviamancine}@discente.ufg.br,danilo.ramos@ufg.br
{valdemarneto,laribeiro}@inf.ufg.br

Resumo. A Controladoria-Geral da União (CGU) enfrenta desafios na gestão e resposta a um volume crescente de recursos relacionadas aos pedidos de acesso à informação. Para abordar este problema, este artigo apresenta o projeto “SARA” (Sistema Automatizado de Resposta a Recursos), uma solução baseada em Processamento de Linguagem Natural que utiliza de Geração Aumentada de Recuperação para identificar recursos e pedidos similares, prever decisões e gerar respostas automatizadas aos recursos. Experimentos preliminares indicam que o projeto SARA tem o potencial de melhorar a eficiência e a velocidade de resposta, sugerindo um mecanismo robusto e escalável para o tratamento de recursos na CGU.

1. Introdução

A Lei Federal nº 12.527/2011, conhecida como Lei de Acesso à Informação (LAI), regulamenta o acesso a informações públicas no Brasil [Brasil 2011]. No âmbito do Poder Executivo Federal, a plataforma Fala.BR facilita esse acesso e é supervisionada pela Controladoria-Geral da União (CGU)¹. Cidadãos têm o direito de solicitar informações públicas, e em caso de negativa, podem recorrer internamente no órgão solicitante, escalando até a CGU se necessário. A CGU deve deliberar sobre esses recursos em até cinco dias, conforme o Artigo 16 da LAI [Brasil 2011]. No entanto, o tempo médio atual de resposta é de 63 dias, devido ao grande volume de recursos ².

Neste contexto, este artigo apresenta o projeto SARA, uma solução baseada em Processamento de Linguagem Natural (PLN) para auxiliar a CGU a cumprir os prazos legais e melhorar a eficiência no tratamento dos recursos. A viabilidade do projeto foi avaliada por meio de uma Prova de Conceito (POC) com dados reais da CGU, oferecendo *insights* sobre o desempenho da solução, além de identificar diretrizes para a continuidade e evolução do projeto.

Este artigo está organizado da seguinte forma: a Seção 2 apresenta o referencial teórico e trabalhos relacionados. A Seção 3 detalha a solução SARA, abordando as técnicas empregadas e a sua arquitetura. Na Seção 4, é apresentado a prova de conceito, incluindo a metodologia de implementação, além das ferramentas e tecnologias utilizadas. A Seção 5 é dedicada à exposição dos experimentos preliminares realizados e à discussão dos resultados obtidos. Por fim, a Seção 6 apresenta as conclusões e os próximos passos para a continuidade e evolução do projeto SARA.

¹<https://www.gov.br/acessoainformacao/pt-br>

²<https://centralpaineis.cgu.gov.br/visualizar/lai>

2. Referencial Teórico e Trabalhos Relacionados

Nesta seção, serão abordados os aspectos técnicos fundamentais para o desenvolvimento do projeto SARA e alguns trabalhos relacionados, com foco especial em *Large Language Model* (LLM) e a geração de índices baseados em *embeddings*, bem como na técnica de *Retrieval Augmented Generation* (RAG).

2.1. Técnicas de Processamento de Linguagem Natural

As técnicas de PLN são essenciais para a análise e geração de textos, permitindo que sistemas automatizados compreendam e respondam a consultas em linguagem natural [Eisenstein 2019]. Nos últimos anos, LLMs, como o GPT-3 e Llama, revolucionaram o campo do PLN [Touvron et al. 2023]. Esses modelos são treinados em grandes corpora de textos e possuem a capacidade de gerar respostas contextualmente relevantes, realizar traduções automáticas, sumarizar textos e outras tarefas [Chang et al. 2024]. No contexto do projeto SARA, os LLMs são utilizados para gerar sugestões de respostas aos recursos.

Outra técnica é a geração de *embeddings*, que envolve a criação de representações vetoriais para textos, permitindo a busca por similaridade entre documentos. Modelos baseados no Sentence-BERT são amplamente utilizados para este propósito, onde *embeddings* são gerados para capturar o contexto semântico dos textos [Reimers and Gurevych 2019]. A avaliação de *embeddings* de texto é discutida de forma abrangente no trabalho de Muennighoff et al. (2023), que introduz o *Massive Text Embedding Benchmark* (MTEB), destacando a diversidade de tarefas e a necessidade de métodos universais de *embeddings* [Muennighoff et al. 2023]. No projeto SARA, a geração de *embeddings* é utilizada para encontrar pedidos e recursos similares, que são posteriormente usados para melhorar a qualidade e relevância das respostas geradas.

Trabalhos relacionados investigaram o impacto do pré-processamento e da representação textual na classificação de documentos de licitações [Brandão et al. 2023], e propuseram uma plataforma para deduplicação de dados em âmbito governamental [Mangaravite et al. 2022]. Esses trabalhos ressaltam a importância de técnicas eficazes de PLN para a melhoria da classificação e integração de dados.

2.2. Retrieval Augmented Generation

RAG é uma técnica que combina métodos de recuperação de informações (IR) com modelos de geração de texto, permitindo ao sistema buscar dados relevantes e gerar respostas de maneira integrada [Ding et al. 2024]. De acordo com Gao et al. (2023), a RAG melhora a precisão das respostas e reduz a alucinação dos modelos, particularmente em tarefas intensivas em conhecimento, ao combinar o conhecimento parametrizado dos LLMs com bases de conhecimento externas não parametrizadas [Gao et al. 2023]. Diferente das abordagens tradicionais de IR, que se concentram na recuperação baseada em consultas, a RAG permite uma interação mais dinâmica e contextual, essencial para lidar com a natureza multifacetada dos pedidos de acesso à informação na CGU. Além disso, o uso de modelos de linguagem superou técnicas tradicionais, como o BM25 [Bonifacio et al. 2022].

O projeto SARA utiliza RAG para identificar e recomendar pedidos e recursos, recuperando informações relevantes de casos passados e gerando respostas adequadas. Essa abordagem permite ao sistema lidar com grandes volumes de dados textuais e mitigar limitações dos LLMs, como alucinações e conhecimento desatualizado, proporcionando uma camada adicional de verificação e atualização das informações.

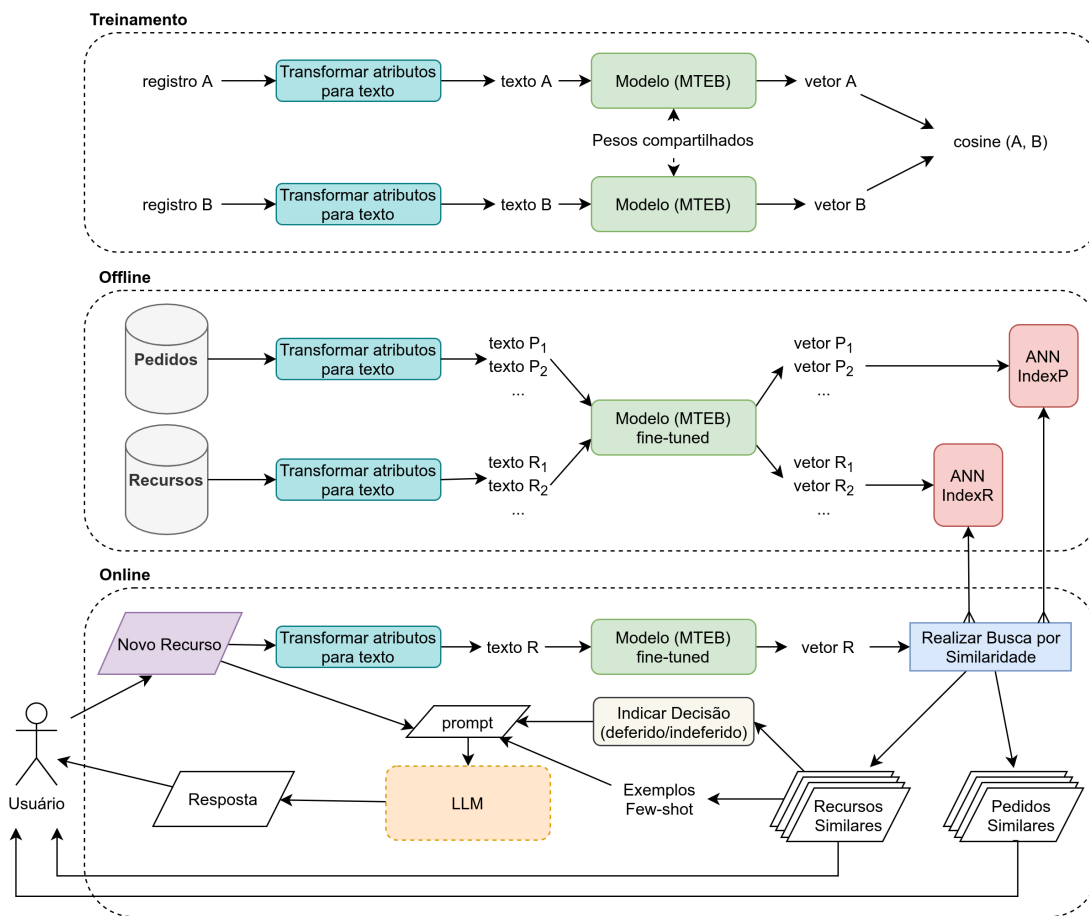


Figura 1. Visão Geral da Solução SARA

3. Solução SARA

A solução SARA utiliza como base a recuperação baseada em *embeddings* e a geração da respostas através de um LLM. As três funcionalidades básicas da solução são: (1) gerar recomendações de pedidos e recursos similares para auxiliar na resposta de um novo recurso; (2) gerar recomendação de decisão para o novo recurso a partir dos recursos similares já respondidos e; (3) gerar recomendação de texto de resposta do recurso com justificativa da decisão. Na Figura 1 é apresentado uma visão geral da arquitetura da solução e abaixo o detalhamento das etapas:

Transformação de atributos para texto: nesta etapa os atributos são convertidos em sequências de texto. É necessário para preparar os dados estruturados de pedidos e recursos para o processamento pelo modelo de linguagem. A seleção cuidadosa dos atributos que melhor representam cada registro é essencial. Após a seleção, a transformação envolve a concatenação desses atributos, que são separados pelo token <SEP>.

Geração de *embeddings*: esta etapa envolve o uso de modelos de linguagem especializados na conversão de texto em vetores semânticos. Desta forma, são gerados *embeddings* onde textos semanticamente similares são representados como vetores próximos no espaço vetorial. Além disso, modelos de última geração utilizam mecanismos de atenção para se concentrar em partes mais informativas, aumentando a eficácia do modelo.

Indexação e Busca: os *embeddings* gerados anteriormente são organizados utilizando algoritmos especializados em indexação, permitindo buscas rápidas e precisas em grandes volumes de dados. A busca por similaridade é então realizada para retornar pedidos e recursos similares a um novo recurso, suportando as funcionalidades de recomendação da decisão e de geração de texto de resposta, além de oferecer ao usuário uma análise mais contextual dos casos, aumentando a eficiência na triagem e resposta aos recursos.

Indicação de decisão para o recurso: nesta etapa, são analisados os recursos similares já respondidos para verificar a probabilidade de diferentes decisões, como deferimento ou indeferimento. De forma simplificada, o SARA calcula os percentuais de cada tipo de decisão entre os recursos similares e recomenda a decisão com o maior percentual. Esta abordagem permite uma análise quantitativa, para sugerir o desfecho mais provável, fundamentando a recomendação em padrões observados em casos anteriores.

Geração de texto da resposta: nesta fase final, o SARA emprega um LLM para gerar automaticamente o texto da resposta com a justificativa da decisão. Este processo envolve fornecer no *prompt* do LLM as informações do novo recurso, a decisão indicada na fase anterior e exemplos de respostas de recursos similares que receberam a mesma decisão. Esta abordagem é baseada na técnica RAG, que potencializa a habilidade do LLM de fornecer respostas contextualizadas e embasadas, extrapolando o conhecimento adquirido durante seu treinamento.

4. Prova de Conceito

A Prova de Conceito (POC) envolveu a implementação completa das etapas da solução SARA utilizando Python e o ambiente Google Colaboratory. Todas as fases do pipeline foram implementadas com o apoio das bibliotecas e modelos de linguagem disponibilizados pelo *Hugging Face*³. O código fonte desta implementação está disponível no repositório *GitHub*⁴.

Dados Utilizados: Os dados para os experimentos foram obtidos da base de recursos e pedidos da LAI, acessíveis através da página de dados abertos da CGU⁵. Foram utilizados os dados de pedidos e recursos de 2013 a 2023, totalizando 780.084 pedidos e 82.124 recursos. A tabela de pedidos contém 21 atributos e a de recursos 17 atributos, conforme dicionário de dados disponibilizado na mesma página de acesso aos dados. Além disso, foram utilizados dados anotados fornecidos pela CGU para testes e avaliação de pedidos similares.

Transformação dos Atributos para Texto: Na POC para o SARA, os atributos selecionados para representar os pedidos foram ‘ResumoSolicitacao’ e ‘DetalhamentoSolicitacao’, e para os recursos foram escolhidos ‘TipoRecurso’ e ‘DescricaoRecurso’. Estes atributos foram escolhidos devido à riqueza de informações que oferecem para a identificação de cada registro.

Geração de *Embeddings*: Para a geração de *embeddings*, foram utilizados modelos pré-treinados indicados no MTEB [Muennighoff et al. 2023], garantindo a eficiência na captura de semântica dos textos. Especificamente, foram utilizados os modelos all-MiniLM-

³<https://huggingface.co/>

⁴<https://github.com/douglasrolins/projeto-sara>

⁵<https://buscalai.cgu.gov.br/DownloadDados/DownloadDados>

L12-v2⁶, all-mpnet-base-v2⁷, intfloat/multilingual-e5-base⁸ e intfloat/multilingual-e5-large⁹. Nesta POC, foram empregados os modelos pré-treinados, sem a realização de uma etapa de *fine-tuning* específica para os dados.

Indexação e Busca: A biblioteca FAISS foi utilizada para a indexação e busca, referência no campo de busca de similaridade em larga escala [Johnson et al. 2019]. O índice empregado foi o IndexFlatIP, que obtém resultados exatos e utiliza o produto interno como função de similaridade.

Indicação de Decisão: Um método dedicado foi desenvolvido para analisar os recursos similares que foram identificados. O objetivo deste método foi verificar qual decisão ocorre com maior frequência entre os casos similares. A decisão identificada sendo mais comum foi indicada ao novo recurso em análise. Esta decisão serviu como base na fase subsequente de geração de resposta, garantindo que a resposta gerada estivesse alinhada com as tendências observadas nos dados.

Geração de Resposta: Para a geração do texto de resposta do recurso, foi empregado o LLM Zephyr 7b-beta¹⁰ [Tunstall et al. 2023]. Devido às limitações de memória no Google Colaboratory, optou-se pela versão quantizada de 4 bits do modelo. O *prompt* fornecido ao LLM incluiu instruções iniciais para realização da tarefa, o tipo e descrição do novo recurso, a decisão indicada na etapa anterior, e duas respostas de recursos similares como exemplos *few-shot*.

5. Experimentos e Resultados

Os experimentos para avaliar a solução SARA na POC focaram na qualidade dos pedidos recomendados, devido à disponibilidade de dados anotados pela CGU. Outras áreas de avaliação, como a qualidade dos recursos recomendados, qualidade da resposta gerada e tempo de execução, serão exploradas em trabalhos futuros. Além disso, para uma visão prática das capacidades do sistema, exemplos de respostas geradas pelo SARA podem ser consultados nos apêndices do relatório técnico disponível no repositório *GitHub*¹¹.

A CGU forneceu dados anotados de pares de pedidos similares, totalizando 2123 pares. Destes, 1113 foram utilizados nos experimentos, sendo o restante excluído por conter pedidos privados não incluídos nos conjuntos de dados públicos. As métricas utilizadas para avaliação foram o *Mean Reciprocal Rank (MRR)* e o *recall@k*. O *MRR* avalia a posição média do primeiro pedido relevante (Definição 1), enquanto o *recall@k* mede a fração de pedidos relevantes nos top-k resultados (Definição 2).

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i} \quad (1)$$

onde $|Q|$ representa o número total de consultas, e rank_i é a posição do primeiro pedido relevante na i -ésima consulta.

⁶<https://huggingface.co/sentence-transformers/all-MiniLM-L12-v2>

⁷<https://huggingface.co/sentence-transformers/all-mpnet-base-v2>

⁸<https://huggingface.co/intfloat/multilingual-e5-base>

⁹<https://huggingface.co/intfloat/multilingual-e5-large>

¹⁰<https://huggingface.co/HuggingFaceH4/zephyr-7b-beta>

¹¹<https://github.com/douglasrolins/projeto-sara>

$$recall@k = \frac{|\{\text{pedidos relevantes encontrados entre os top-k}\}|}{|\{\text{pedidos relevantes totais}\}|} \quad (2)$$

Os resultados são apresentados na Tabela 1 e indicam superioridade do modelo “multilingual-e5-base”, com um recall de 0.712 para $k = 100$. As diferenças no tempo de geração dos *embeddings* para cada modelo são apresentadas e detalhadas na Tabela 1. Para a execução desta tarefa, utilizou-se uma GPU Nvidia A100 com 40Gb de VRAM no ambiente do Google Colaboratory.

Os resultados de *MRR* e *recall@k*, especialmente para k menores, indicam que a solução não recomendou a maioria dos pares similares anotados, possivelmente devido à quantidade limitada de dados anotados em comparação ao tamanho total do conjunto de dados. Outro ponto é a presença frequente de pedidos com conteúdos idênticos no campo ‘DetalhamentoSolicitacao’, sugerindo a necessidade de um tratamento diferenciado para reduzir redundâncias nos resultados. O *fine-tuning* dos modelos com dados específicos de pedidos não foi realizado nesta POC, sendo uma recomendação para futuras avaliações.

Model	Dim.	Tempo Encoding	MRR	recall@5	recall@10	recall@20	recall@50	recall@100
all-MiniLM-L12-v2	384	9m25s	0.365	0.305	0.395	0.465	0.549	0.627
all-mpnet-base-v2	768	33m39s	0.361	0.292	0.378	0.443	0.537	0.610
multilingual-e5-base	768	30m14s	0.396	0.365	0.454	0.526	0.633	0.712
multilingual-e5-large	1024	39m28s	0.397	0.333	0.421	0.496	0.600	0.662

Tabela 1. Resultados das recomendações de pedidos

6. Conclusão

Este trabalho apresentou o projeto SARA, um sistema para auxiliar a CGU no tratamento e resposta automatizada a recursos sobre pedidos de acesso à informação. A solução integra técnicas de geração de *embeddings*, indexação para buscas por similaridade e geração automática de textos por meio de um LLM. Experimentos preliminares avaliaram a qualidade das recomendações de pedidos, fornecendo *insights* para a evolução do projeto e potencial implantação em um ambiente real.

Como futuras direções, serão realizadas comparações com *baselines* tradicionais, como métodos convencionais de IR. Recomenda-se também estudar combinações de atributos para otimização dos vetores representativos e realizar *fine-tuning* dos modelos para melhor adaptação aos dados. A geração de mais dados anotados é fundamental para aprimorar a avaliação da solução, assim como a análise de índices aproximados para indexação, especialmente para implantação em ambientes com grandes volumes de dados. Além disso, a implementação de um produto mínimo viável é importante para testar a solução em um cenário prático.

Espera-se que a solução SARA melhore a eficiência operacional da CGU e agregue qualidade ao processo decisório, fornecendo recomendações e respostas contextualizadas a partir de uma análise automatizada de dados históricos.

Referências

- Bonifacio, L., Abonizio, H., Fadaee, M., and Nogueira, R. (2022). Inpars: Unsupervised dataset generation for information retrieval. In *Proceedings of the 45th International ACM SIGIR, SIGIR '22*, page 2387–2392, New York, NY, USA.
- Brandão, M., Silva, M., Oliveira, G., Hott, H., Lacerda, A., and Pappa, G. (2023). Impacto do Pré-processamento e Representação Textual na Classificação de Documentos de Licitações. In *Anais do XXXVIII Simpósio Brasileiro de Bancos de Dados*, pages 102–114, Porto Alegre, RS, Brasil. SBC.
- Brasil (2011). Lei nº 12.527, de 18 de Novembro de 2011. Lei de Acesso à Informação.
- Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., Ye, W., Zhang, Y., Chang, Y., Yu, P. S., Yang, Q., and Xie, X. (2024). A Survey on Evaluation of Large Language Models. *ACM Trans. Intell. Syst. Technol.*
- Ding, Y., Fan, W., Ning, L., Wang, S., Li, H., Yin, D., Chua, T.-S., and Li, Q. (2024). A Survey on RAG Meets LLMs: Towards Retrieval-Augmented Large Language Models. *arXiv preprint arXiv:2405.06211*.
- Eisenstein, J. (2019). *Introduction to Natural Language Processing*. MIT Press.
- Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., and Wang, H. (2023). Retrieval-Augmented Generation for Large Language Models: A Survey. *arXiv preprint arXiv:2312.10997*.
- Johnson, J., Douze, M., and Jégou, H. (2019). Billion-Scale Similarity Search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.
- Mangaravite, V., Carvalho, M., Cantelli, L., Ponce, L., Campoi, B., Nunes, G., Laender, A., and Gonçalves, M. (2022). DedupeGov: Uma Plataforma para Integração de Grandes Volumes de Dados de Pessoas Físicas e Jurídicas em Âmbito Governamental. In *Anais do XXXVII SBBDD*, pages 90–102, Porto Alegre, RS, Brasil. SBC.
- Muennighoff, N., Tazi, N., Magne, L., and Reimers, N. (2023). MTEB: Massive Text Embedding Benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia. Association for Computational Linguistics.
- Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. ACL.
- Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. (2023). Llama: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971*.
- Tunstall, L., Beeching, E., Lambert, N., Rajani, N., Rasul, K., Belkada, Y., Huang, S., von Werra, L., Fourrier, C., Habib, N., et al. (2023). Zephyr: Direct Distillation of LM Alignment. *arXiv preprint arXiv:2310.16944*.