Relatório Final

Gabriel Cassol Bach, Bruno Emanuel Zenatti

8 de dezembro de 2025

1 Introdução

Este trabalho apresenta o Tech Discovers, sistema desenvolvido para identificar padrões, tendências e inovações tecnológicas em patentes. Nesta seção, será apresentado o funcionamento geral do sistema, fundamentado na abordagem de Recuperação Aumentada por Geração (RAG), bem como os dados utilizados para sua construção. Por fim, serão expostas as perguntas de pesquisa que norteiam a condução do estudo e a análise dos resultados ao longo do relatório.

A relevância do sistema emerge da necessidade de automatizar parte do processo de análise de patentes, que tradicionalmente exige leituras extensas e comparações manuais. Ao organizar informações e fornecer respostas com base em documentos relevantes, o Tech Discovers contribui para tornar o processo de investigação tecnológica mais ágil, preciso e acessível a diferentes perfis de usuários.

Os dados utilizados neste trabalho são provenientes do Harvard USPTO Dataset (HUPD) [SCLH22], que inclui aproximadamente 4,5 milhões de solicitações de patentes feitas nos Estados Unidos entre 2011 e 2016. Para fins de análise, selecionou-se apenas o ano de 2016, resultando em um subconjunto de cerca de 373 mil patentes

A fim de compreender e aplicar as técnicas estudadas na disciplina de Processamento de Linguagem Natural, optou-se por desenvolver um sistema baseado em RAG [LPP+20]. Essa abordagem foi escolhida por permitir integrar recuperação de informações com geração automática de respostas, possibilitando buscar documentos relevantes e sintetizar seus conteúdos de forma eficiente.

A partir desse contexto, este trabalho busca responder às seguintes perguntas de pesquisa:

1. **Quais estratégias de recuperação são mais eficazes para encontrar patentes relevantes?**
2. **Como o RAG facilita tarefas de análise de patentes, como identificação de antecedentes e inovações?**
3. **Quais são as limitações das buscas utilizando o RAG?**

Essas questões direcionam o desenvolvimento do sistema e orientam a análise dos resultados obtidos, permitindo identificar tanto os benefícios quanto os desafios associados ao uso de RAG no contexto de análises de patentes.

2 Metodologia

Nesta seção será apresentada a metodologia adotada para construir o sistema de análise de patentes baseado em RAG. Descreveremos o fluxo de processamento desde a preparação dos dados até a geração das respostas, incluindo a extração de palavras-chave, a vetorização das patentes, a construção do índice vetorial e a definição das regras de *prompting* e formatação das respostas.

2.1 Extração de Informação para Vetorização

Para cada registro de patente extraem-se, como campos primários, o título e o resumo. Esses campos são armazenados em um cache de dados que conserva o texto bruto necessário para, posteriormente, apresentar contexto ao usuário.

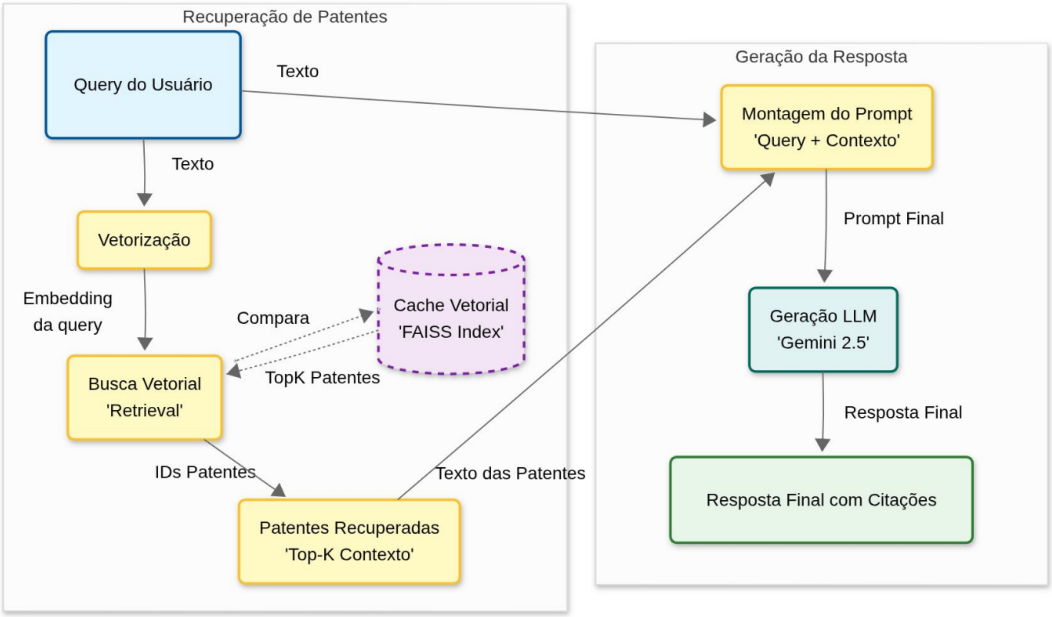

Figura 1: Esquema Geral do Sistema

2.2 Geração de Palavras-chave

A partir do resumo, são identificadas expressões relevantes (nome de produtos, componentes, métodos, domínios técnicos). Essas entidades são normalizadas e filtradas (remoção de stopwords, termos genéricos e duplicatas) gerando uma lista de palavras-chave por documento. As palavras-chave são anexadas ao título e ao resumo, formando a representação textual final que será vetorizada. A lógica da geração de palavras-chave pode ser melhor compreendida pela Figura 2.

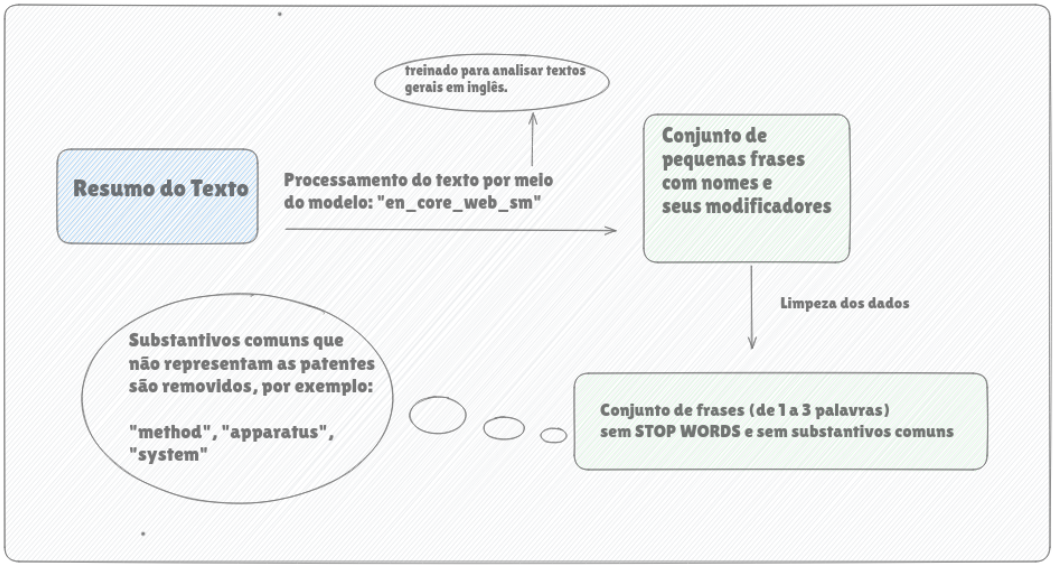

Figura 2: Geração de Palavras-Chave

2.3 Criação do Sistema de Recuperação (FAISS)

O sistema de vetorização transforma a entrada unificada (título + resumo + palavras-chave) em um embedding numérico via um modelo de SentenceTransformer. Os vetores são salvos em cache *(.npy)* para evitar recomputação. Em seguida, constrói-se um índice FAISS (FAISSIndex) [JDJ17] com esses vetores normalizados, usando similaridade de cosseno como métrica. Esse índice permite comparar rapidamente uma pergunta vetorizada com todo o conjunto de patentes.

2.4 Recuperação das Patentes

Diante de uma consulta, a frase informada pelo usuário é transformada em vetor e comparada com o índice existente. Obtêm-se então as k patentes mais similares, acompanhadas de suas pontuações. Como os índices retornados permitem o mapeamento direto ao conjunto original armazenado, recupera-se o texto integral associado a cada patente encontrada, incluindo resumo, título e palavras-chave. Esse material constitui a base de conhecimento contextual usada na resposta. Importante também destacar que, para o modelo `bge-large` [XLZM23], é recomendada a utilização de uma string de instrução antes da *query* de busca: `"Represent this sentence for searching relevant passages: "`, a qual foi utilizada de forma a melhorar a qualidade do *retrieval*.

2.5 Geração do Texto para o Usuário via LLM

Com os documentos recuperados, o módulo que encapsula as funções do RAG (recuperação e geração) gera o prompt que será enviado ao modelo de linguagem. Para cada documento filtrado, extrai-se o texto original, a lista de entidades e o score. Esses blocos são concatenados, formando o contexto que será usado no *prompt* e enviado para a LLM (Gemini 2.5 Flash - [Dee25]).

Regras explícitas do *prompt*:

1. Usar apenas a informação presente no contexto;
2. Responder *insufficient evidence* se não houver suporte completo da resposta pelo contexto;
3. Citar os IDs das patentes para cada afirmação factual;
4. Ao final da resposta, listar exatamente as patentes referenciadas e o *score* da patente (valor de similaridade entre query e patente);
5. Se nenhum documento passa por um limiar de *score*, retorna *insufficient evidence*.

Esse desenho garante que a LLM sintetize respostas diretamente amparadas pelos trechos recuperados, mantendo rastreabilidade (IDs e scores) e prevenindo inserção de informação externa. O uso combinado de geração de palavras-chave, vetorização consistente, indexação FAISS e regras de prompting oferece um pipeline reprodutível e adequado para análise automatizada de patentes.

2.6 Sistema Web para Interação com o RAG

Além do processamento das patentes, foi desenvolvido um sistema web simples - apresentado na Figura 3 - para permitir que o usuário utilize o modelo RAG de forma prática. O sistema conta com uma interface onde o usuário pode inserir perguntas em linguagem natural e visualizar os resultados retornados. Quando uma consulta é enviada, o backend executa a recuperação vetorial, identifica as patentes mais relevantes e gera uma resposta baseada no conteúdo encontrado. Assim, a aplicação torna o uso do sistema mais acessível, sem exigir conhecimento técnico direto em programação.

3 Avaliação e Resultados

Para validar a eficácia do sistema, conduziu-se uma avaliação qualitativa manual baseada em um conjunto de 10 perguntas diversificadas, cobrindo diferentes domínios tecnológicos (e.g., saúde, engenharia elétrica, redes de computadores). O objetivo foi mensurar não apenas a precisão da recuperação de documentos, mas também a capacidade de síntese e fidelidade da LLM.

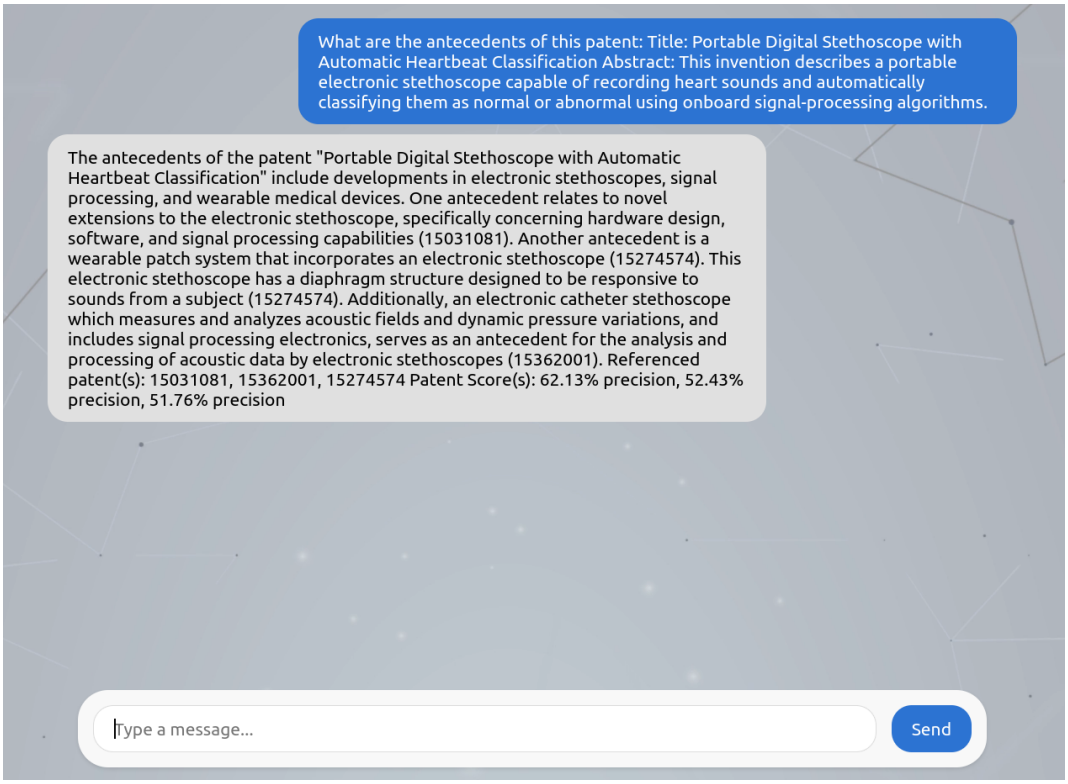

Figura 3: Exemplo do Funcionamento do Sistema

3.1 Metodologia de Avaliação e Critérios

A avaliação comparou duas configurações de parâmetros para identificar o impacto das estratégias de recuperação na qualidade final da resposta:

- **Configuração A:** Modelo de *embedding* `all-MiniLM-L6-v2`, [Hug21] com *Top-K*=3.
- **Configuração B:** Modelo de *embedding* `BAAI/bge-large-en-v1.5`, [XLZM23] com *Top-K*=5.

Os critérios de sucesso definidos foram:

1. **Capacidade de Recuperação (Retrieval):** O sistema recuperou patentes relevantes para a consulta?
2. **Fidelidade e Alucinação:** A resposta foi gerada estritamente com base no contexto fornecido? As citações correspondem aos IDs corretos?
3. **Coerência:** A resposta é fluida e responde diretamente à intenção do usuário?

3.2 Análise dos Resultados Quantitativos

A Tabela 1 apresenta as notas atribuídas (escala 0-10) para as respostas geradas em cada configuração. Observou-se que a Configuração B (`bge-large` [XLZM23], Top-K=5) apresentou, em média, uma consistência superior, especialmente em domínios onde a terminologia é mais densa.

3.3 Discussão dos Parâmetros

A análise dos logs de execução permitiu inferir comportamentos críticos sobre os parâmetros do sistema RAG:

- **Score Threshold:** As análises indicaram que um limiar de similaridade (*score*) inferior a 50% recupera patentes com baixa relação semântica, introduzindo ruído. Em contrapartida, limiares superiores a 60%, embora aumentem a similaridade mínima das patentes recuperadas, tendem a

descartar documentos relevantes. Portanto, o ponto de equilíbrio ideal foi identificado em torno de 50%. Vale ressaltar que a própria LLM atua como um filtro secundário de relevância: ao se deparar com contextos pouco relevantes (mesmo que superem o limiar de recuperação), o modelo é instruído a responder "insufficient evidence". Essa característica de segurança permite configurar o *threshold* de recuperação em valores mais baixos, como 0.1%, sem comprometer a confiabilidade da resposta final.

- **Janela de Contexto (Top-K):** A expansão de *Top-K*=3 para *Top-K*=5 provou ser benéfica. Ao contrário do esperado, o aumento no volume de texto não causou "confusão"na LLM (Gemini 2.5 Flash). Pelo contrário, forneceu mais informações, quando haviam, para uma resposta mais completa, como observado no teste sobre "Machine Learning e NLP", onde a nota subiu de 5.0 para 9.0 devido à melhor integração dos temas.
- **Comportamento em Consultas Vagas:** No teste 9 ("Antecedentes de patentes relacionadas a lentes de câmera"), a consulta mal formulada levou o sistema a não encontrar evidências suficientes. O comportamento da LLM foi correto ao retornar "insufficient evidence"ou notas baixas de recuperação, demonstrando que o *prompt* de sistema foi eficaz em evitar alucinações quando o contexto era pobre.

Um desafio recorrente identificado nos logs foi a presença de patentes duplicadas ou com pequenas variações de texto no *Top-K*, o que consome a janela de contexto com informação redundante, limitando a diversidade da resposta final.

Tabela 1: Comparação de Desempenho entre Configurações por Query

#	**Query do Usuário**	**Nota (Conf. A)**	**Nota (Conf. B)**
1	Analyze antecedents of this patent: The present invention relates to a medicinal composition useful for the treatment of diabetes in a human patient...	9.0	7.0
2	Give me ideas of computer networks innovations	7.0	10.0
3	Can you find alternatives to this patent: "Biodegradable polymers: New types of polymers for packaging or medical devices..."	9.0	8.0
4	What are the innovations presented in the drone area? Related to autonomous robots	7.0	9.0
5	Most important electrical machinery and energy creations	10.0	7.5
6	Inventions and discoveries related to machine learning, information retrieval and NLP	5.0	9.0
7	Wearable devices related to cardiovascular health, monitoring and improving exercises	10.0	9.0
8	Show me different innovations about e-cigarette	7.0	10.0
9	Give me antecedents of patents related to camera lens	6.0	8.0
10	Camera's lens. Photography	10.0	8.0

4 Trabalhos Futuros

Com base nas limitações observadas durante a avaliação, delineiam-se as seguintes direções para a evolução do sistema:

1. **Refinamento de Consultas:** A qualidade da recuperação é dependente da formulação da

pergunta. Propõe-se o desenvolvimento de um módulo intermediário que reformule a consulta do usuário ou a implementação de sugestões de *templates* na interface para guiar o usuário na criação de perguntas mais assertivas.
2. **Limpeza de Dados:** Foi notado que diversas patentes recuperadas são idênticas. Uma etapa de pré-processamento para agrupar ou remover duplicatas no índice vetorial aumentaria a variabilidade de informações fornecidas à LLM.
3. **Aprimoramento do Embedding:** Testes com modelos fine-tunados especificamente para textos técnicos/jurídicos, poderiam elevar a qualidade do *retrieval*.

5 Conclusão

Este trabalho apresentou o desenvolvimento e avaliação do *Tech Discovers*, um sistema baseado em RAG para análise automatizada de patentes. Através da integração de busca vetorial densa com modelos de linguagem de grande porte, foi possível validar uma ferramenta capaz de acelerar a extração de conhecimento técnico. Retomando as perguntas de pesquisa formuladas na introdução, conclui-se que:

1. **Quais estratégias de recuperação são mais eficazes?** O uso de modelos de *embedding* mais robustos (como o `bge-large` [XLZM23]) combinados com um número maior de documentos recuperados (*Top-K*=5) mostrou-se superior a modelos mais leves.
2. **Como o RAG facilita tarefas de análise de patentes?** O sistema demonstrou alta capacidade de sintetizar informações complexas, como a identificação de inovações em drones autônomos. A arquitetura permitiu que usuários obtivessem respostas diretas e fundamentadas, com rastreabilidade garantida pelas citações dos IDs das patentes.
3. **Quais são as limitações das buscas utilizando o RAG?** A principal limitação reside na sensibilidade à qualidade da consulta do usuário e na redundância dos dados. Consultas vagas resultam em recuperação pobre, e a presença de documentos duplicados no conjunto de dados reduz a eficiência da janela de contexto da LLM.

Em suma, o sistema atingiu seu objetivo de identificar padrões e inovações, oferecendo uma interface acessível para exploração de grandes bases de propriedade intelectual.

Referências

[Dee25] Google DeepMind. Gemini 2.5 flash model. https://deepmind.google/technologies/gemini/, 2025. Acesso em: 06 dez. 2025.

[Hug21] Hugging Face. Sentence-Transformers model: all-MiniLM-L6-v2. https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2, 2021. Acessado em: 07 dez. 2025.

[JDJ17] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Faiss: A library for efficient similarity search and clustering of dense vectors. https://github.com/facebookresearch/faiss, 2017. Acesso em: 2025.

[LPP+20] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Madian Kulkarni, Mike Lewis, Wen-tau Yih, and Sebastian Riedel. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Proceedings of NeurIPS*, 2020.

[SCLH22] Mirac Suzgun, Yiming Chen, Percy Liang, and Tatsunori B. Hashimoto. Harvard uspto patent dataset (hupd). https://github.com/suzgunmirac/hupd, 2022.

[XLZM23] Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. C-pack: Packaged resources to advance general chinese embedding. https://huggingface.co/BAAI/bge-large-en-v1.5, 2023.