UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA

João Marcos Lyra Vieira

Restauração Automática de Hierarquias em Documentos Corporativos: Uma comparação entre métodos de vetorização e similaridade

Recife

2025

João Marcos Lyra Vieira

Restauração Automática de Hierarquias em Documentos Corporativos: Uma comparação entre métodos de vetorização e similaridade

Trabalho apresentado ao Programa de Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Área de Concentração: Processamento de Linguagem Natural e Recuperação de Informação

Orientador (a): Filipe Calegário

Recife

2025

Ficha de identificação da obra elaborada pelo autor,
através do programa de geração automática do SIB/UFPE

Vieira, João Marcos Lyra.
Restauração automática de hierarquias em documentos corporativos: uma comparação entre métodos de vetorização e similaridade / João Marcos Lyra Vieira. - Recife, 2025.
49 p, tab.

Orientador(a): Filipe Carlos de Albuquerque Calegario
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Pernambuco, Centro de Informática, Ciências da Computação - Bacharelado, 2025.
Inclui referências.

1. Restauração de documentos. 2. SBert. 3. Similaridade de cosseno. 4. Embeddings. I. Calegario, Filipe Carlos de Albuquerque. (Orientação). II. Título.

000 CDD (22.ed.)

João Marcos Lyra Vieira

Restauração automática de hierarquias em documentos corporativos: uma comparação entre métodos de vetorização e similaridade

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em 2025 da Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de bacharel em Ciência da Computação.

Aprovado em: *05*/*08*/2025

BANCA EXAMINADORA

_____

Prof. Dr. Filipe Calegário (Orientador)

Universidade Federal de Pernambuco

_____

Prof. Dr. Ricardo Prudêncio (Examinador Interno)

Universidade Federal de Pernambuco

Dedico este trabalho ao meu pai, à minha mãe e à minha irmã, por todo o apoio o e pelo amor incondicional. Estendo esta dedicatória a todos os amigos que cultivei ao longo do curso, pelos momentos de apoio, aprendizado e cumplicidade.

AGRADECIMENTOS

Agradeço, em primeiro lugar, ao meu orientador, Prof. Dr. Filipe Calegário, pela orientação atenta, pelos ensinamentos e pela confiança depositada neste trabalho.

À Universidade Federal de Pernambuco e ao Centro de Informática, pelo ambiente acadêmico de excelência e pelos recursos disponibilizados.

Aos meus pais e à minha irmã, pelo apoio constante, incentivo e carinho que tornaram possível cada etapa desta jornada.

Aos amigos que fiz durante o curso, pelas discussões, momentos de descontração e colaboração que enriqueceram minha formação pessoal e profissional.

Por fim, agradeço a todos que, de alguma forma, contribuíram para a realização deste trabalho e de toda a minha jornada acadêmica.

RESUMO

"O crescimento acelerado da documentação digital nas empresas tem aumentado significativamente o desafio de organizar e recuperar informações. Um exemplo comum dessa dificuldade é a perda das relações hierárquicas entre documentos corporativos devido à fragmentação do armazenamento em múltiplas bases de dados. Neste trabalho, investigou-se o problema da restauração automática das relações hierárquicas de pai–filho entre documentos corporativos, a partir de seus títulos e resumos. Inicialmente, implementou-se uma abordagem baseada em Bag of Words e similaridade de cosseno (limiar = 0,60), que apresentou alta precisão, porém baixa cobertura. Em seguida, conduziram-se experimentos comparativos envolvendo cinco técnicas de vetorização (TF-IDF, Word2Vec, FastText, SBert e OpenAI embeddings) combinadas com três métricas de similaridade (cosseno, distância euclidiana e Jaccard) sobre uma base de aproximadamente 9 000 artigos científicos. Os resultados mostram que os embeddings SBert com similaridade de cosseno alcançaram a maior média de Similaridade Temática (0,8619), seguidos pelos embeddings OpenAI (0,8537). A comparação com a versão inicial do projeto evidenciou que SBert duplicou ou triplicou a cobertura de pares pai–filho, mantendo mais de 85% de correspondência com os pares originais. Conclui-se que a combinação SBert + similaridade de cosseno oferece o melhor equilíbrio entre cobertura, consistência e custo, sendo recomendada para aplicações práticas.

Palavras-chaves: restauração de documentos. SBert. similaridade de cosseno. embeddings.

ABSTRACT

The rapid growth of digital documentation within companies has significantly increased the challenge of organizing and retrieving information. A common illustration of this difficulty is the loss of hierarchical parent–child relationships among corporate documents due to storage fragmentation across multiple databases. In this work, we investigated the problem of automatically restoring hierarchical parent–child relationships between corporate documents based on their titles and abstracts. Initially, we implemented an approach based on a Bag-of-Words representation and cosine similarity (threshold $= 0.60$), which achieved high precision but low coverage. We then conducted comparative experiments involving five vectorization techniques (TF-IDF, Word2Vec, FastText, SBert, and OpenAI embeddings) combined with three similarity metrics (cosine, Euclidean distance, and Jaccard) on a dataset of approximately 9,000 scientific articles. The results show that SBert embeddings with cosine similarity achieved the highest mean Thematic Similarity (0.8619), followed by OpenAI embeddings (0.8537). Comparison with the initial version of the project demonstrated that SBert doubled or tripled the coverage of parent–child pairs while maintaining over 85% agreement with the original pairs. We conclude that the combination of SBert and cosine similarity offers the best balance of coverage, consistency, and cost, making it recommended for practical applications.

Keywords: document restoration. SBert. cosine similarity. embeddings.

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
BoW	Bag of Words
CBOW	Continuous Bag of Words
TF	Term Frequency
IDF	Inverse Document Frequency
TF-IDF	Term Frequency–Inverse Document Frequency
ST	Similaridade Temática
IT	Interseção Temática
UT	União Temática
PLN	Processamento de Linguagem Natural
NLP	Natural Language Processing
GenAI	Inteligência Artificial Generativa
BERT	Bidirectional Encoder Representations from Transformers
SBERT	Sentence-BERT
GPT-4o	Generative Pre-trained Transformer versão 4 otimizada

SUMÁRIO

1 INTRODUÇÃO

A gestão eficiente e organizada de documentos corporativos é um elemento fundamental para o bom funcionamento e a governança de organizações modernas. Empresas frequentemente estruturam seus documentos em hierarquias claras, onde documentos de nível superior, como políticas institucionais, sustentam documentos derivados ou de nível inferior, como guias operacionais e procedimentos específicos. No entanto, algumas empresas constroem e armazenam seus documentos em diversos repositórios e com a iminente necessidade de unificação da base de documentos, a preocupação em ter uma ferramenta que possa encontrar os relacionamentos entre documentos começa a ganhar cada vez mais força.

Nesse contexto, surgiu um desafio específico em uma empresa onde realizei meu estágio: as relações entre documentos corporativos de diversas bases não existiam, dificultando a recuperação eficiente da informação e comprometendo a gestão do conhecimento organizacional. Para solucionar esse problema, foi desenvolvido um sistema em Python que estabelecia relações hierárquicas por meio de técnicas básicas de vetorização textual sintática e similaridade de cosseno, permitindo identificar, dentre diversos documentos possíveis, o documento-pai mais adequado para cada documento-filho. Porém, com um limiar de 60% estabelecido, o número de documentos-filho sem pai identificado ainda seguiu muito alto, tornando necessário o estudo de técnicas que tornem essa busca por relacionamentos mais eficaz.

1.1 MOTIVAÇÃO

A motivação deste trabalho reside na necessidade prática e urgente de estabelecer e documentar as relações entre documentos corporativos semanticamente relacionados, visto que sua ausência prejudica diretamente processos internos, gera retrabalho e aumenta o risco de falhas operacionais. Além disso, existe uma oportunidade acadêmica relevante na avaliação e comparação detalhada de técnicas avançadas de processamento de linguagem natural aplicadas ao contexto corporativo, contribuindo tanto para o meio acadêmico quanto para o setor produtivo.

1.2 OBJETIVOS

O objetivo principal deste trabalho é realizar uma avaliação comparativa das técnicas mais utilizadas na literatura para a tarefa de encontrar similaridades semânticas entre documentos. Especificamente, busca-se:

- Avaliar comparativamente cinco técnicas diferentes para a representação textual (TF-IDF, Word2Vec, FastText, Embeddings SBert e Embeddings OpenAI);
- Avaliar três algoritmos de similaridade (Similaridade de Cosseno, Distância Euclidiana e Similaridade de Jaccard);
- Desenvolver e validar uma métrica própria denominada "Similaridade Temática", construída por meio da classificação automática dos documentos em seis categorias distintas utilizando inteligência artificial generativa (GenAI);
- Analisar os resultados das melhores técnicas no conjunto de dados reais, a fim de entender os benefícios práticos trazidos por este estudo.

Espera-se que os resultados obtidos nesta pesquisa possam oferecer não apenas uma solução robusta e eficaz para o problema inicialmente identificado na empresa, mas também contribuir para futuras aplicações em contextos similares, proporcionando avanços significativos na gestão inteligente de documentos corporativos.

2 FUNDAMENTAÇÃO TEÓRICA

A fundamentação teórica deste trabalho abrange três áreas essenciais ao entendimento das técnicas propostas: representação vetorial de documentos, algoritmos de similaridade e métodos avançados de processamento de linguagem natural.

2.1 REPRESENTAÇÃO VETORIAL DE DOCUMENTOS

A representação textual por meio de vetores numéricos é uma etapa fundamental para realizar tarefas eficazes em processamento de linguagem natural (PLN) (ASUDANI; NAGWANI; SINGH, 2023; AMAN, 2023). Entre as técnicas mais relevantes utilizadas neste contexto estão TF-IDF, Word2Vec, FastText, Embeddings SBert e Embeddings OpenAI. Conforme a Tabela 1, é possível analisar o comparativo entre as técnicas antes de entendê-las individualmente.

Tabela 1 – Comparação entre modelos de vetorização textual

Modelo	**Tipo de Representação**	**Considera Contexto?**	**Trabalha com Sentenças?**	**Lida com Palavras Raras?**	**Aplicações Comuns**
TF-IDF	Matriz esparsa	Não	Não	Não	Classificação, recuperação de informação
Word2Vec	Vetores densos	Sim	Não	Não	NLP, análise semântica, similaridade de palavras
FastText	Vetores densos	Sim	Não	Sim	Modelos de linguagem, tarefas multilíngues
Sentence Transformer	Vetores densos	Sim	Sim	Sim	Busca semântica, análise de similaridade
Embeddings OpenA	Vetores densos	Sim	Sim	Sim	NLP, recuperação de informação, análise de texto

Fonte: Elaborada pelo autor.

2.1.1 Bag of Words (BoW)

A técnica Bag of Words é uma abordagem simples e eficaz para representar textos como vetores numéricos, considerando apenas a frequência das palavras em cada documento, ignorando completamente a ordem das palavras e a estrutura gramatical. Cada documento é transformado em um vetor que contabiliza quantas vezes cada palavra aparece nele, resultando em uma matriz esparsa. Exemplo prático: consiste em extrair as 1.000 palavras mais frequentes de uma coleção de notícias e representar cada artigo por quantas vezes cada palavra aparece nele.

Embora eficaz para tarefas básicas de classificação e recuperação de informação, esta técnica não captura nuances semânticas ou contextuais dos termos utilizados, o que pode ser uma limitação significativa para aplicações mais complexas. Apesar de sua relevância histórica e simplicidade, esta técnica não foi incluída na Tabela 1 por não ter sido utilizada diretamente nesta versão final do trabalho.

2.1.2 TF-IDF (Term Frequency-Inverse Document Frequency)

Essa técnica estatística avalia a importância relativa das palavras em documentos individuais em relação ao corpus inteiro. O peso de uma palavra é proporcional à sua frequência no documento específico (Term Frequency - TF) e inversamente proporcional à frequência da palavra no corpus inteiro (Inverse Document Frequency - IDF). TF-IDF é amplamente utilizado por sua simplicidade e eficiência na captura de termos importantes para caracterizar documentos (AMAN, 2023).

2.1.3 Word2Vec

Um modelo baseado em redes neurais artificiais, especificamente nas arquiteturas Continuous Bag of Words (CBOW) e Skip-Gram. Word2Vec aprende a representação distribuída (embeddings) de palavras ao analisar grandes quantidades de texto, capturando relações semânticas e sintáticas (ASUDANI; NAGWANI; SINGH, 2023; SIMPLR, 2023). Embeddings obtidos pelo Word2Vec permitem representar palavras como vetores densos que preservam a semântica e são úteis em diversas aplicações como classificação, agrupamento e similaridade textual.

2.1.4 FastText

Uma extensão do modelo Word2Vec, também baseada em redes neurais superficiais, que considera a composição interna das palavras usando n-grams (subpalavras). Essa técnica oferece vantagens significativas em situações com vocabulários amplos ou palavras raras, garantindo representações robustas mesmo para termos desconhecidos durante o treinamento inicial (ASUDANI; NAGWANI; SINGH, 2023; SIMPLR, 2023).

2.1.5 Embeddings SBert (Sentence-BERT)

Baseada no modelo de linguagem pré-treinado BERT (Bidirectional Encoder Representations from Transformers), o SBert gera representações semânticas precisas em nível de frases e parágrafos completos (REIMERS; GUREVYCH, 2019). Ao contrário de abordagens tradicionais, o SBert captura não só o significado individual das palavras, mas também o contexto completo em que aparecem, tornando-se especialmente eficaz em tarefas complexas de similaridade textual e recuperação de informações.

2.1.6 Embeddings OpenAI

Embeddings provenientes de modelos generativos avançados, como GPT (Generative Pre-trained Transformer), treinados em grandes volumes de texto da internet. Esses embeddings destacam-se pela capacidade de representar conteúdos textuais com profundidade e riqueza semântica, sendo ideais para tarefas que exigem compreensão sofisticada do contexto, incluindo classificação, análise semântica e recuperação de documentos. Para este trabalho, foi utilizado o modelo text-embedding-3-small, que apresenta um custo-benefício razoável comparado com as demais opções fornecidas pela OpenAI (OpenAI, 2022).

2.2 ALGORITMOS DE SIMILARIDADE

Após a representação dos textos em vetores, aplicam-se algoritmos que medem quantitativamente o grau de semelhança entre esses vetores (VIDAL, 2021; SINGH, 2024). Os principais algoritmos considerados neste estudo são:

- Similaridade de Cosseno: Amplamente utilizada em PLN, mede o ângulo entre vetores em um espaço multidimensional, representando o grau de alinhamento semântico entre documentos. É particularmente eficaz com representações densas e esparsas, sendo frequentemente escolhida por sua capacidade de capturar similaridades semânticas sutis.

- Distância Euclidiana: Baseia-se na distância geométrica entre pontos em um espaço multidimensional. Embora menos sensível a relações semânticas sutis em representações não-normalizadas, torna-se equivalente à similaridade de cosseno quando os vetores são previamente normalizados, situação comum em representações semânticas modernas.

- Similaridade de Jaccard: Mede a interseção sobre a união entre conjuntos de palavras ou tokens de dois documentos. Seu uso é mais comum em contextos onde a comparação direta de conjuntos discretos é desejada, como na detecção de similaridade em listas de palavras-chave, textos curtos ou contextos específicos, como análise de plágio.

2.3 MÉTODOS AVANÇADOS DE PLN E AVALIAÇÃO

Para complementar as abordagens tradicionais, este trabalho emprega métodos avançados utilizando inteligência artificial generativa (GenAI) e grandes modelos de linguagem (LLMs). Modelos como GPT-4o da OpenAI, ao serem alimentados com engenharia sofisticada de prompts, permitem classificar documentos em categorias temáticas definidas previamente, além de fornecer avaliações qualitativas detalhadas sobre a semelhança textual (WANG; AL., 2023; BAYTAS; RUEDIGER, 2024; TULLY; AL., 2024).

Essas capacidades permitiram a criação de uma métrica inovadora, a "Similaridade Temática", que quantifica a semelhança entre documentos com base na presença de categorias temáticas comuns atribuídas automaticamente pelos modelos generativos. A utilização adicional da avaliação qualitativa assistida por LLMs proporciona insights enriquecedores e detalhados sobre o porquê das similaridades ou divergências observadas, garantindo resultados robustos e bem fundamentados.

Dessa forma, a combinação dessas técnicas avançadas visa garantir uma análise abrangente, detalhada e rigorosa das técnicas de representação e similaridade, contribuindo significativamente para soluções eficientes e confiáveis no campo de processamento de documentos.

3 METODOLOGIA

Para alcançar o objetivo proposto neste trabalho, a metodologia foi dividida em etapas claramente definidas: preparação dos dados, experimentos realizados e avaliação dos resultados obtidos.

3.1 PREPARAÇÃO DOS DADOS

A base de dados utilizada para os experimentos é composta por títulos e resumos de cerca de nove mil artigos científicos (Kaggle user arashnic, 2025). Essa base foi originalmente criada para um desafio de Machine Learning visando à classificação multi-temática dos artigos nos seguintes tópicos: Computer Science, Mathematics, Physics, Statistics, Quantitative Biology e Quantitative Finance.

Para validação das técnicas empregadas por meio da métrica da Similaridade Temática, foi realizada uma classificação automatizada de artigos científicos em larga escala utilizando a API da OpenAI com o modelo GPT-4o mini. Para isso, foi desenvolvido um prompt (WANG; AL., 2023), no qual foi definido o modelo como um "especialista em taxonomia arXiv", além de especificar sete categorias e definir um limiar de confiança de 80% para redução de falsos positivos. Foi incorporada uma breve cadeia de pensamento passo a passo para que o modelo identifique palavras-chave, resuma o conteúdo e avalie cada área antes de gerar a saída em JSON estritamente formatado, oferecendo assim uma base consistente para as etapas subsequentes. A seguir, apresentamos o prompt completo utilizado em nossos experimentos.

```
1  You are an expert classifier of scientific articles, familiar with the arXiv
       taxonomy.

3  Given a TITLE and ABSTRACT of an article, assign it to each of the following six
       categories:
   - Computer Science
5  - Mathematics
   - Physics
7  - Statistics
   - Quantitative Biology
9  - Quantitative Finance

11 Rules:
   1. Use only the TITLE and ABSTRACT.
13 2. Only mark "Yes" if you are >= 80% confident the article is primarily about
```

```
    that field; otherwise "No".
3. Before answering, think step by step:
   a. Identify the main keywords and concepts in the TITLE.
   b. Summarize the ABSTRACT in one sentence.
   c. For each category, ask yourself: "Does this summary strongly relate to the
       core topics of CATEGORY?"
   d. Judge whether that relation is >= 80% confident.
4. At the end of the reasoning, the final result should be an output **exactly**
    in JSON (without text or extra formatting). It is very important that the
    final output is in the requested format and without any type of extra
    formatting.

### Example

TITLE: "A Novel Algorithm for Distributed Graph Coloring"
ABSTRACT: "We introduce a distributed algorithm for graph coloring that runs in
    logarithmic time..."
**Think:**
1. Keywords: distributed, graph coloring, algorithm
2. Summary: Introduces a fast distributed graph-coloring method.
3. Does this summary strongly relate to the core topics of Computer Science? -> "
    Yes"
4. Does this summary strongly relate to the core topics of Mathematics? -> "Yes"
5. Does this summary strongly relate to the core topics of Physics? -> "No"
6. Does this summary strongly relate to the core topics of Statistics? -> "Yes"
7. Does this summary strongly relate to the core topics of Quantitative Biology?
    -> "No"
8. Does this summary strongly relate to the core topics of Quantitative Finance?
    -> "No"
OUTPUT:
{
  "Computer Science": "Yes",
  "Mathematics": "Yes",
  "Physics": "No",
  "Statistics": "Yes",
  "Quantitative Biology": "No",
  "Quantitative Finance": "No"
}

### Now classify this article:

TITLE: {title}
ABSTRACT: {abstract}

**Think step by step** and then produce the JSON.
OUTPUT:
```

3.2 EXPERIMENTOS

Os experimentos foram estruturados em torno de diversas combinações entre técnicas de vetorização e algoritmos de similaridade.

Para as As técnicas de vetorização testadas foram:

- TF-IDF
- Word2Vec
- FastText
- Embeddings SBert (all-MiniLM-L6-v2)
- Embeddings OpenAI (text-embedding-3-small)

Cada técnica foi implementada utilizando bibliotecas Python adequadas: TF-IDF com Scikit-learn, Word2Vec e FastText utilizando Gensim, SBert com a biblioteca SentenceTransformers e Embeddings da OpenAI acessados via API da empresa.

Já os algoritmos de similaridade empregados foram:

- Similaridade de Cosseno
- Distância Euclidiana
- Similaridade de Jaccard

Cabe destacar que, devido à sua natureza, a Similaridade de Jaccard foi adaptada especificamente para vetores densos (Word2Vec, FastText, SBert e Embeddings OpenAI), enquanto sua versão tradicional foi aplicada somente ao TF-IDF.

O experimento consistiu em, para cada artigo da base e cada combinação técnica-algoritmo, encontrar os três artigos mais similares, formando assim três pares de documentos para análise posterior.

3.3 AVALIAÇÃO

A avaliação dos resultados foi realizada através da métrica Similaridade Temática (ST), especialmente desenvolvida para este estudo. A ST é definida como a razão entre a interseção

e a união dos tópicos atribuídos aos documentos comparados, representada pela fórmula:

$$ST(A, B) = \frac{|IT(A, B)|}{|UT(A, B)|}$$

Onde:

- IT(A, B) representa o número de tópicos em comum entre dois artigos.
- UT(A, B) é o total de tópicos presentes em ao menos um dos artigos.

A utilização dessa métrica se dá pelo contexto hierárquico e corporativo a que se deseja aplicar as técnicas com melhores resultados. Neste contexto, os documentos-filho precisam conter todos os tópicos abordados pelo respectivo documento-pai, dessa forma, ao comparar documentos com a ST, podemos metrificar essa relação.

Para cada combinação de técnicas de vetorização e algoritmos de similaridade, calculou-se a média da Similaridade Temática para os três documentos mais semelhantes de cada artigo e, por fim, obteve-se uma média geral representativa do desempenho das técnicas.

A validação da métrica de Similaridade Temática foi complementada e avaliada por uma análise subjetiva assistida pelo modelo GPT-4o, que consistiu em analisar pares de artigos com diferentes níveis de Similaridade Temática e realizar uma avaliação manual da similaridade desses artigos, a fim de conferir se a métrica criada de fato traduziu a semelhança entre os documentos.

Foram definidos três níveis de similaridade temática para a avaliação subjetiva assistida:

- Fortemente similares: valores entre 0,8 e 1,0
- Similaridade média: valores entre 0,3 e 0,7
- Baixa ou nenhuma similaridade: valores inferiores a 0,3

A métrica foi validada com sucesso por meio de 15 amostras aleatórias. Dentre os pares avaliados, houve concordância nas análises subjetivas em 14 casos, o que corresponde a aproximadamente 93% de acerto. A única discordância ocorreu em um par classificado, pela análise subjetiva, como de similaridade média, porém a similaridade temática deste par foi de 0,25, indicando uma similaridade baixa. Isso indica que, por uma diferença de apenas 0,05 no índice de similaridade temática, não foi possível alcançar 100% de concordância.

4 RESULTADOS E DISCUSSÃO

Nesta seção, apresentam-se de forma sistemática os resultados obtidos para cada combinação de técnica de vetorização e métrica de similaridade, seguida de uma análise comparativa e interpretação dos achados. Utiliza-se como métrica principal a *Similaridade Temática (ST)*, calculada para os três artigos mais similares a cada item da base testada. Em seguida, discutem-se implicações práticas dos resultados, suas limitações, além de testes com dados corporativos reais e possíveis desdobramentos para uso nesse contexto.

4.1 INTRODUÇÃO AOS DADOS CONSOLIDADOS

Antes de detalhar cada técnica, é importante mostrar a *tabela consolidada das médias de ST* para todas as combinações (vetorização $\times$ métrica). Essa tabela resume, para cada técnica de vetorização, a média de Similaridade Temática obtida ao usar as três métricas de similaridade (Cosseno, Euclidiana e Jaccard).

Tabela 2 – Médias de Similaridade Temática (ST) por Técnica de Vetorização e Métrica de Similaridade

Técnica de Vetorização	Métrica de Similaridade	Média ST
TF-IDF	Similaridade de Cosseno	0,6987
	Distância Euclidiana	0,7435
	Similaridade de Jaccard	0,7013
Word2Vec	Similaridade de Cosseno	0,6876
	Distância Euclidiana	0,7125
	Jaccard ponderado	0,7378
FastText	Similaridade de Cosseno	0,6629
	Distância Euclidiana	0,6958
	Jaccard ponderado	0,7452
SBert	Similaridade de Cosseno	0,8619
	Distância Euclidiana	0,8619
	Jaccard ponderado	0,7478
OpenAI	Similaridade de Cosseno	0,8537
	Distância Euclidiana	0,8537
	Jaccard ponderado	0,7315

Ao observar as médias consolidadas de ST, percebe-se que as técnicas baseadas em Trans-

formers (SBert e embeddings OpenAI) obtêm valores idênticos para as métricas de Similaridade de Cosseno e Distância Euclidiana (0,8619 e 0,8537, respectivamente), o que reflete a forte correlação entre essas medidas quando aplicadas a vetores densos normalizados. Por outro lado, nas técnicas tradicionais de vetorização (TF-IDF, Word2Vec e FastText), a Distância Euclidiana tende a apresentar médias ligeiramente superiores às da Similaridade de Cosseno, enquanto a Similaridade de Jaccard assume valores intermediários, com pequenas variações conforme cada modelo. O maior valor absoluto de ST foi alcançado por SBert (0,8619), seguido de perto pelos embeddings OpenAI (0,8537), evidenciando o ganho de desempenho proporcionado pelos modelos de linguagem avançados. Em contraste, o menor desempenho médio foi observado para FastText com Similaridade de Cosseno (0,6629); contudo, ao empregar o Jaccard ponderado nesse mesmo modelo, obteve-se um resultado competitivo (0,7452), sugerindo que a escolha da métrica pode atenuar limitações inerentes à representação por subpalavras.

Em virtude dessa consolidação, esta seção está estruturada em dois blocos principais. Primeiro, na Seção 4.2 apresentamos os resultados detalhados para cada técnica de vetorização, incluindo a distribuição de ST (percentis, desvios-padrão) e a correlação entre as métricas. Em seguida, na Seção 4.3 comparamos de forma geral as técnicas, discutindo as combinações mais promissoras, os trade-offs entre desempenho e custo computacional e as implicações para a tarefa de restauração automática de hierarquias documentais.

4.2 RESULTADOS POR TÉCNICA DE VETORIZAÇÃO

Nesta seção, cada subseção apresenta:

1. **Breve descrição do procedimento** para cálculo da ST na respectiva técnica;
2. **Principais estatísticas** (média, desvio-padrão, percentis) dos valores de ST obtidos para Cosseno, Euclidiana e Jaccard;
3. **Correlação estatística** entre as métricas (coeficiente de correlação de Pearson);
4. **Discussão específica** dos achados dessa técnica.

4.2.1 TF-IDF

4.2.1.1 Procedimento para cálculo da ST:

1. Representam-se os e resumos dos artigos como vetores TF-IDF (termos $\times$ documentos).
2. Para cada artigo "âncora", calculam-se as similaridades (Cosseno, Distância Euclidiana e Jaccard) frente aos demais itens da base.
3. Selecionam-se, para cada artigo, os três mais similares segundo a métrica em questão.
4. Para cada par (A, B), extrai-se a lista de "tópicos" de A e de B (utilizados no experimento de classificação temática) e computa-se:

$$\mathrm{ST}(A, B) \;=\; \frac{|IT(A, B)|}{|UT(A, B)|}.$$

5. A média dos três valores de ST por artigo é registrada, e então calculada a média geral de ST para toda a técnica.

4.2.1.2 Principais estatísticas (Valores médios de ST):

Tabela 3 – Estatísticas de ST para TF-IDF

Métrica	**Média ST**	**Desvio-padrão**	**P25**	**P75**
Cosseno	0,6987	0,09	0,62	0,77
Dist. Euclidiana	0,7435	0,08	0,67	0,81
Jaccard (binar.)	0,7013	0,10	0,60	0,79

4.2.1.3 Correlação entre métricas (Pearson):

- $r(\text{Cosseno}, \text{Euclidiana}) = 0{,}87$
- $r(\text{Cosseno}, \text{Jaccard}) = 0{,}75$
- $r(\text{Euclidiana}, \text{Jaccard}) = 0{,}78$

4.2.1.4 Discussão (TF-IDF):

Na análise baseada em TF-IDF, observa-se que a Distância Euclidiana supera, em média, em aproximadamente seis pontos percentuais a Similaridade de Cosseno (0,7435 vs. 0,6987). Esse comportamento pode ser atribuído à característica esparsa dos vetores TF-IDF, em que documentos concisos apresentam distâncias mais expressivas, enquanto a métrica de cosseno aplica uma normalização pela norma do vetor, atenuando tais discrepâncias. Quando a Similaridade de Jaccard é empregada em vetores TF-IDF binarizados — considerando apenas a presença ou ausência de termos — ela produz resultados muito próximos à métrica de cosseno (0,7013 vs. 0,6987), confirmando que, nesse cenário, a maior parte da informação relevante está na simples sobreposição de palavras-chave.

O elevado coeficiente de correlação de Pearson ($r = 0{,}87$) entre as medições de cosseno e euclidiana indica que ambas as métricas mantêm ordenações similares dos pares de documentos, embora a fórmula euclidiana penalize menos as variações entre vetores esparsos. Em síntese, o TF-IDF apresenta desempenho intermediário: é confiável para comparar títulos e resumos com boa coincidência de termos, mas fica aquém das técnicas baseadas em embeddings densos, sobretudo quando se exige sensibilidade semântica mais refinada.

4.2.2 Word2Vec

4.2.2.1 Procedimento para cálculo da ST:

1. Constrói-se um modelo Word2Vec sobre todo o corpus de títulos e resumos.
2. Cada título de artigo é representado pela média (ou soma ponderada) dos vetores de suas palavras-chave.
3. Calculam-se similaridades (Cosseno, Euclidiana e *Jaccard ponderado*) entre os vetores médios dos documentos.
4. Para o *Jaccard ponderado*, utiliza-se a fórmula adaptada para vetores densos:

$$\text{Jaccard}_{\text{ponderado}}(u, v) \;=\; \frac{\sum_i \min(u_i, v_i)}{\sum_i \max(u_i, v_i)}.$$

5. Calcula-se a ST para os três artigos mais similares e, em seguida, a média geral de ST.

4.2.2.2 *Principais estatísticas (Valores médios de ST):*

Tabela 4 – Estatísticas de ST para Word2Vec

Métrica	**Média ST**	**Desvio-padrão**	**P25**	**P75**
Cosseno	0,6876	0,12	0,57	0,81
Dist. Euclidiana	0,7125	0,11	0,60	0,83
Jaccard ponderado	0,7378	0,10	0,63	0,85

4.2.2.3 *Correlação entre métricas (Pearson):*

- $r(\text{Cosseno}, \text{Euclidiana}) = 0{,}81$
- $r(\text{Cosseno}, \text{Jaccard ponderado}) = 0{,}69$
- $r(\text{Euclidiana}, \text{Jaccard ponderado}) = 0{,}74$

4.2.2.4 *Discussão (Word2Vec):*

Na aplicação da técnica Word2Vec, observa-se que a Similaridade de Jaccard ponderado alcançou o melhor desempenho (0,7378), sugerindo que, ao considerar o peso relativo de cada dimensão dos embeddings, essa métrica capta de forma mais eficaz a sobreposição temática entre documentos. As medições por Cosseno e Distância Euclidiana apresentaram médias inferiores em relação ao Jaccard ponderado, mas ainda exibiram correlação moderada (r = 0,81). Dentre elas, a métrica euclidiana ofereceu ligeira vantagem em magnitude (0,7125 vs. 0,6876), refletindo maior sensibilidade às variações nos vetores médios.

Em termos gerais, o Word2Vec ficou abaixo dos modelos SBert e OpenAI no que diz respeito à média de Similaridade Temática, mas superou o TF-IDF ao empregar Jaccard ponderado (0,7378 $>$ 0,7013). Esse achado reforça que embeddings derivados de redes neurais conseguem capturar relações semânticas ausentes na representação baseada em contagem de TF-IDF.

4.2.3 FastText

4.2.3.1 Procedimento para cálculo da ST:

1. Utiliza-se um modelo pré-treinado de FastText (palavras e subpalavras).
2. Cada título e resumo de artigo é representado pela média dos embeddings de cada palavra.
3. Calculam-se Cosseno, Euclidiana e Jaccard ponderado entre vetores médios de documentos.
4. Computa-se ST para os três artigos mais similares e, depois, a média geral.

Principais estatísticas (Valores médios de ST):

Tabela 5 – Estatísticas de ST para FastText

Métrica	Média ST	Desvio-padrão	P25	P75
Cosseno	0,6629	0,13	0,52	0,79
Dist. Euclidiana	0,6958	0,12	0,58	0,82
Jaccard ponderado	0,7452	0,11	0,63	0,87

4.2.3.2 Correlação entre métricas (Pearson):

- $r(\text{Cosseno}, \text{Euclidiana}) = 0{,}79$
- $r(\text{Cosseno}, \text{Jaccard ponderado}) = 0{,}65$
- $r(\text{Euclidiana}, \text{Jaccard ponderado}) = 0{,}71$

4.2.3.3 Discussão (FastText):

Na análise com FastText, a métrica de Jaccard ponderado destacou-se como a que obteve a maior média de Similaridade Temática (0,7452), o que indica que, ao lidar com embeddings de subpalavras, atribuir pesos às dimensões maximiza a captura de semelhanças temáticas entre documentos. Em contraste, a Similaridade de Cosseno apresentou o resultado mais baixo

de todas as combinações avaliadas (0,6629), sugerindo que vetores oriundos de subpalavras podem gerar maior variabilidade em documentos de vocabulário próximo, tornando o cosseno menos estável sob essas condições. A Distância Euclidiana, por sua vez, ficou em patamar intermediário (0,6958), superior ao cosseno mas aquém dos valores observados para Word2Vec e TF-IDF, o que reforça a ideia de que o ganho semântico do FastText só se reflete plenamente quando uma métrica sensível à sobreposição de características, como o Jaccard ponderado, é empregada.

4.2.4 SBert

4.2.4.1 Procedimento para cálculo da ST:

1. Carrega-se o modelo pré-treinado SBert (`all-MiniLM-L6-v2`).
2. Cada conjunto de título e resumo de artigo é codificado como vetor denso de alta dimensão (384 dimensões).
3. Calculam-se similaridades por Cosseno, Distância Euclidiana e Jaccard ponderado entre vetores densos.
4. Computa-se ST para os três artigos mais similares e, depois, a média geral.

4.2.4.2 Principais estatísticas (Valores médios de ST):

Tabela 6 – Estatísticas de ST para SBert

Métrica	**Média ST**	**Desvio-padrão**	**P25**	**P75**
Cosseno	0,8619	0,07	0,81	0,92
Dist. Euclidiana	0,8619	0,07	0,81	0,92
Jaccard ponderado	0,7478	0,09	0,65	0,84

4.2.4.3 Correlação entre métricas (Pearson):

- $r(\text{Cosseno}, \text{Euclidiana}) = 0{,}99$

- $r(\text{Cosseno, Jaccard ponderado}) = 0{,}62$
- $r(\text{Euclidiana, Jaccard ponderado}) = 0{,}62$

4.2.4.4 Discussão (SBert):

Na aplicação de SBert, observa-se que as métricas de Similaridade de Cosseno e Distância Euclidiana apresentam valores idênticos (0,8619), pois os embeddings gerados pelo modelo são normalizados por padrão (REIMERS; GUREVYCH, 2019). Dessa forma, temos

$$\text{Cosseno}(u, v) = u \cdot v \quad \text{e} \quad \|u - v\|_2^2 = 2 - 2\,(u \cdot v),$$

o que implica que ordenar pares de documentos por cosseno ou por distância euclidiana produz a mesma topologia de similaridade.

O valor médio de ST de 0,8619 alcançado por SBert (tanto em Cosseno quanto em Euclidiana) é o mais elevado entre todas as técnicas testadas, evidenciando a capacidade desse modelo em capturar relações semânticas profundas entre títulos e resumos, mesmo na ausência de termos idênticos. Já a Similaridade de Jaccard ponderado para SBert (0,7478), embora inferior ao produto escalar, ainda supera confortavelmente as melhores combinações de TF-IDF, Word2Vec e FastText. Isso sugere que, apesar de útil para mensurar sobreposição de componentes, o produto escalar reflete de forma mais fidedigna a semântica implícita nos embeddings densos do SBert.

Por fim, o coeficiente de correlação de Pearson extremamente alto (r = 0,99) entre Cosseno e Euclidiana confirma que, em espaços de vetores normalizados, essas duas medidas são praticamente intercambiáveis, reiterando a robustez de SBert para a tarefa de avaliação de similaridade temática.

4.2.5 Embeddings OpenAI (text-embedding-3-small)

4.2.5.1 Procedimento para cálculo da ST:

1. Para cada título de artigo, chama-se a API OpenAI (`text-embedding-3-small`) para obter vetores densos de 1536 dimensões.
2. Realizam-se as mesmas comparações de Cosseno, Euclidiana e Jaccard ponderado.

3. Buscam-se os três artigos mais similares por métrica e calcula-se ST conforme descrito anteriormente.

4.2.5.2 Principais estatísticas (Valores médios de ST):

Tabela 7 – Estatísticas de ST para Embeddings OpenAI

Métrica	**Média ST**	**Desvio-padrão**	**P25**	**P75**
Cosseno	0,8537	0,08	0,79	0,91
Dist. Euclidiana	0,8537	0,08	0,79	0,91
Jaccard ponderado	0,7315	0,10	0,61	0,83

4.2.5.3 Correlação entre métricas (Pearson):

- $r(\text{Cosseno}, \text{Euclidiana}) = 0{,}99$
- $r(\text{Cosseno}, \text{Jaccard ponderado}) = 0{,}58$
- $r(\text{Euclidiana}, \text{Jaccard ponderado}) = 0{,}58$

4.2.5.4 Discussão (OpenAI):

Nos embeddings gerados pela API OpenAI, observa-se novamente que as métricas de Similaridade de Cosseno e Distância Euclidiana retornam valores idênticos (0,8537), em virtude da normalização interna dos vetores adotada pelo modelo (OpenAI, 2022). Esse comportamento reforça a mesma equivalência topológica já vista em SBert, em que

$$\text{Cosseno}(u, v) = u \cdot v \quad \text{e} \quad \|u - v\|_2^2 = 2 - 2\,(u \cdot v),$$

A média de ST de 0,8537 para Cosseno/Euclidiana, embora ligeiramente inferior ao pico de SBert (0,8619), ainda supera todas as demais técnicas tradicionais e baseadas em Word2Vec ou FastText, confirmando a eficácia dos embeddings OpenAI em capturar nuances semânticas. Por sua vez, a Similaridade de Jaccard ponderado atinge 0,7315 — valor um pouco abaixo do SBert (0,7478) — o que sugere que, para esses vetores densos, o produto escalar (Cosseno) tende a refletir com maior fidelidade a similaridade temática do que a simples sobreposição de componentes ponderados.

Em termos de custo-benefício, embora OpenAI e SBert apresentem desempenhos comparáveis, vale destacar que SBert roda localmente de forma gratuita, enquanto o uso da API da OpenAI implica cobrança por requisição, o que pode influenciar a escolha em cenários de grande escala ou orçamentos restritos.

4.3 COMPARAÇÃO GERAL ENTRE TÉCNICAS

4.3.1 Visão Consolidada das Médias de ST

Reproduzindo, de forma resumida, as médias de ST obtidas por técnica e métrica (conforme Tabela 2):

Tabela 8 – Resumo das Médias de ST (por Técnica e Métrica)

Técnica	**Cosseno**	**Euclidiana**	**Jaccard (c/ ou p.)**
TF-IDF	0,6987	0,7435	0,7013
Word2Vec	0,6876	0,7125	0,7378
FastText	0,6629	0,6958	0,7452
SBert	**0,8619**	**0,8619**	0,7478
OpenAI	**0,8537**	**0,8537**	0,7315

Os principais pontos a serem destacados:

1. **SBert (Cosseno/Euclidiana = 0,8619)** foi a técnica que alcançou a maior média de ST, seguida de **OpenAI (0,8537)**.

2. **TF-IDF (0,6987) e Word2Vec (0,6876)** tiveram as menores médias quando usadas com Cosseno, mas apresentaram ganho de quase 4 p.p. ao empregar a Distância Euclidiana.

3. **FastText** teve o menor valor de Cosseno (0,6629), mas seu Jaccard ponderado de 0,7452 superou ligeiramente o Jaccard ponderado de Word2Vec (0,7378), sugerindo que, para FastText, a métrica Jaccard ponderado é a mais apropriada.

4.3.2 Interpretação dos Resultados

Os achados apontam, em primeiro lugar, para o desempenho superior das técnicas baseadas em transformers (**SBert** e **OpenAI**), cujas médias de ST diferem em apenas 0,0082. Isso comprova que embeddings obtidos a partir de arquiteturas transformer, afinadas para semântica de sentenças, são mais eficazes na captura de similaridade temática entre títulos e resumos de artigos do que representações por vetores de palavras isoladas. Além disso, a equivalência dos valores de Cosseno e Euclidiana em ambos os casos reforça que, em espaços vetoriais normalizados, qualquer uma dessas duas métricas pode ser usada sem perda significativa de informação.

Em seguida, nota-se a importância da métrica de *Jaccard ponderado* quando aplicável a embeddings simples, como Word2Vec e FastText. Nesses modelos, Jaccard ponderado superou consistentemente Coseno e Euclidiana (Word2Vec: 0,7378 vs. 0,6876/0,7125; FastText: 0,7452 vs. 0,6629/0,6958), sugerindo que, ao avaliar vetores densos não normalizados, a medida de "sobreposição" de componentes captura melhor similaridade conceitual do que comparações angulares ou distanciamento euclidiano.

Quanto ao trade-off entre custo e desempenho, o SBert se destaca por ser executado localmente e sem custos adicionais, mostrando-se ideal em cenários com orçamentos limitados para chamadas de API. Os embeddings da OpenAI, por sua vez, entregam desempenho quase idêntico, mas implicam cobrança por token ou requisição, o que pode pesar em projetos de grande escala ou com uso intensivo. Técnicas como Word2Vec e FastText exigem etapa de treinamento (ou uso de modelos pré-treinados gratuitos) e atenção especial ao cálculo de Jaccard ponderado, enquanto o TF-IDF é trivial de implementar, porém apresenta desempenho inferior.

Outro ponto relevante é a consistência das medições: SBert e OpenAI apresentaram menor dispersão de ST (desvio-padrão $\approx 0{,}07$–$0{,}08$) nas métricas de Cosseno/Euclidiana, indicando maior robustez na atribuição de similaridade temática a diferentes pares de documentos. Em contraste, TF-IDF, Word2Vec e FastText mostraram dispersão mais elevada ($\approx 0{,}09$–$0{,}13$), refletindo maior sensibilidade a variações de vocabulário ou comprimento dos textos.

Por fim, para a aplicação prática de restauração automática de hierarquias documentais, recomenda-se priorizar combinações que equilibrem alta ST, consistência e viabilidade operacional: — **SBert + Cosseno/Euclidiana**: combinação ideal, com maior média de ST, alta consistência e sem custo de API; — **OpenAI + Cosseno/Euclidiana**: segunda op-

ção, especialmente se houver necessidade de explorar versões mais robustas (por exemplo, `text-embedding-3-large`), ciente dos custos por requisição; — **FastText/Word2Vec + Jaccard ponderado**: alternativa para cenários intermediários, quando já houver modelo local pré-treinado, aceitando desempenho intermediário ($\approx 0,74$ vs. $\approx 0,86$).

4.3.3 Limitações dos Experimentos

Apesar dos resultados promissores, este estudo apresenta algumas limitações importantes. A avaliação comparativa apoiou-se exclusivamente na Similaridade Temática (ST), adotando uma abordagem subjetiva com o auxílio de um especialista em gerenciamento de documentos de uma empresa privada, mas sem incorporar outras métricas ou métodos alternativos de similaridade textual, o que pode restringir a visão sobre o desempenho relativo das técnicas testadas. Além disso, o SBert foi utilizado em sua versão pré-treinada padrão, sem nenhum tipo de fine-tuning — considerando que tanto os dados de teste quanto os documentos corporativos usados na validação eram majoritariamente em inglês, linguagem padrão do modelo pré-treinado — o que pode comprometer a captura das nuances semânticas próprias do contexto corporativo dos documentos utilizados. Outra limitação refere-se ao escopo dos dados analisados: optou-se por trabalhar apenas com títulos e resumos de artigos, excluindo o texto completo. Essa escolha decorreu da estratégia de expor o mínimo de informações sensíveis possível, em razão da segurança e da criticidade dos documentos corporativos envolvidos, mesmo ao se utilizar modelos locais. Por fim, embora o SBert permita execução local gratuita, o processamento de cerca de 9 000 documentos requer recursos computacionais elevados; por outro lado, a utilização da API da OpenAI, apesar de oferecer modelos de alta qualidade, implica custos por tokens de entrada e saída, o que pode limitar a escalabilidade do método em ambientes com infraestrutura restrita. Reconhecer essas limitações é fundamental para orientar melhorias metodológicas futuras, tais como a inclusão de múltiplas métricas de avaliação e a extensão da análise para o texto completo dos documentos.

4.3.4 Interpretações e Implicações

1. **Por que embeddings densos capturam melhor a hierarquia temática?** Títulos e resumos de documentos corporativos nem sempre compartilham palavras-chave exatas; muitas vezes, dois documentos relacionados utilizam vocabulário diferente para tratar

temas semelhantes. Modelos como SBert e OpenAI aprendem representações semânticas de nível sentencial, de modo que vetores de frases muito parecidas (mesmo sem termos idênticos) permanecem próximos no espaço vetorial. Assim, essas técnicas tendem a apresentar ST mais alta, pois, mesmo sem palavras em comum, há sobreposição semântica profunda.

2. **Limitações observadas:**

   - Em casos de *textos muito curtos* (2–3 palavras), até SBert/OpenAI podem sofrer para distinguir nuances temáticas, eventualmente gerando falso-positivo.
   - Para documentos de *conteúdo extenso*, utilizou-se apenas o título e o resumo no experimento; recomenda-se, em trabalhos futuros, testar a composição de embeddings trechos principais, não apenas o título e o resumo.
   - Embora SBert seja gratuito para uso local, o processamento em lotes de 9 000 artigos exige capacidade computacional razoável (GPU ou CPU multicore), o que pode aumentar o tempo de inferência.

3. **Recomendações de uso:**

   - *SBert + Cosseno/Euclidiana*: primeiro candidato a ser adotado em produção, por fornecer alta ST, consistência e custo zero.
   - *OpenAI + Cosseno/Euclidiana*: segunda opção, especialmente se houver interesse em explorar versões mais robustas (ex. `text-embedding-3-large`), ciente do custo por requisição.
   - *FastText/Word2Vec + Jaccard ponderado*: alternativas quando se deseja driblar limitações de hardware (podem ser rodados localmente em CPU), aceitando desempenho intermediário.
   - Evitar usar *TF-IDF* em casos de texto sujeito a sinonímia ou variações linguísticas, pois tende a negligenciar relações semânticas profundas. Contudo, pode funcionar bem para hierarquias baseadas em vocabulário técnico muito padronizado (ex. nomenclaturas de políticas de empresa).

4.3.5 Sugestões para Trabalhos Futuros

1. **Combinação de métricas:** Testar combinações ponderadas, por exemplo:

$$\alpha \cdot \text{Cosseno} \; + \; (1 - \alpha) \cdot \text{Jaccard}.$$

   Avaliar se uma fusão das métricas melhora a precisão de ST, especialmente em Word2Vec e FastText.

2. **Utilização de multimodalidade:**

   - Em vez de usar apenas o título e resumo, incorporar análises de texto completo para enriquecer a similaridade temática. Técnicas de IA Generativa em escala podem ser utilizadas para encontrar trechos-chave em cada artigo.
   - Combinar embeddings SBert do título com embeddings de documento completo (ex. embeddings gerados por modelos *Longformer*) e verificar ganho de ST em hierarquias mais complexas.

3. **Otimização de parâmetros:**

   - Para SBert, explorar versões mais leves ou quantizações (ex. *SBert multi-qa-MiniLM*) para reduzir tempo de inferência sem perda significativa de ST.
   - Em Word2Vec e FastText, testar diferentes dimensões de embedding, tamanho de janela e critérios de frequência mínima para verificar impacto na ST.

4.3.6 Conclusão dos experimentos

Os experimentos evidenciam que, para a tarefa de restauração automática de relações hierárquicas entre documentos, *SBert (ou, alternativamente, OpenAI text-embedding)* combinados com *Cosseno* ou *Distância Euclidiana* fornecem as melhores taxas médias de Similaridade Temática, alcançando cerca de **0,86**. Técnicas baseadas em vetores de palavras isoladas (Word2Vec, FastText) ainda são úteis, mas requerem a adoção de *Jaccard ponderado* para chegar a resultados competitivos ($\approx$ 0,74), o que, na prática, implica um trabalho adicional de implementação. *TF-IDF*, embora simples e de rápida execução, atinge ST média de 0,70 (com Cosseno), indicando limitações em cenários com vocabulário não padronizado.

5 COMPARAÇÃO ENTRE VERSÕES DO PROJETO

Nesta seção, comparam-se quantitativamente os resultados obtidos na *versão inicial* do projeto (que empregava Bag of Words + Similaridade de Cosseno, com limiar de 60%) e a *versão final* baseada em SBert + Similaridade de Cosseno. O objetivo é avaliar o ganho de cobertura (número de documentos-filho cujo documento-pai foi identificado) e a validade dos pares reconhecidos em ambas as abordagens, considerando que ambas operaram apenas sobre os *títulos* dos documentos.

5.1 CONTEXTUALIZAÇÃO DA VERSÃO INICIAL E LIMITAÇÕES

A **versão inicial** do projeto utilizava:

- Vetorização via *Bag of Words* (BOW) nos *títulos* dos documentos.
- Cálculo de Similaridade de Cosseno entre vetores BOW.
- Limiar fixo de similaridade: $\theta = 0{,}60$.

Nesse esquema, sempre que dois títulos apresentavam uma similaridade de cosseno maior que $0{,}60$, considerava-se que havia relação *pai–filho* entre os documentos correspondentes. Em casos em que havia mais de um possível pai com similaridade maior que o limiar, então o documento com maior similaridade era considerado o pai. Contudo, observou-se que:

1. **Baixa Cobertura de Pais Encontrados:** Com o limiar de 60%, poucos documentos-filho conseguiram identificar qualquer documento-pai, pois muitos títulos apresentavam vocabulário distinto ou vocabulário semelhante porém com variações de ordem e frequência que não atingiam θ.
2. **Dependência Sintática:** A abordagem BOW + Cosseno, por operar estritamente sobre contagem de termos, não captava semântica quando o vocabulário variava, ainda que houvesse forte relação temática.
3. **Uso Exclusivo de Títulos:** A extração do texto completo dos documentos demandaria esforço manual e envolvia questões de segurança da informação (propriedade intelectual, confidencialidade). Por isso, decidiu-se trabalhar apenas com títulos, que podiam ser extraídos de forma mais direta e consistente.

Apesar da baixa cobertura, sempre que o cosseno entre dois títulos ultrapassava 0,60, a *avaliadora especialista* (analista de conteúdo) confirmava, de maneira quase unânime, a existência de relação pai–filho real. Esse achado sugere que a versão inicial, embora conservadora, produzia pares de alta precisão (baixa taxa de falsos positivos), porém com *alta taxa de falsos negativos*.

5.2 CONFIGURAÇÃO DA VERSÃO FINAL (SBERT + COSSENO)

A **versão final** substituiu a vetorização BOW por embeddings SBert (modelo `all-MiniLM-L6-v2`), mantendo o cálculo de Similaridade de Cosseno. As principais características dessa implementação foram:

- Vetorização semântica via SBert sobre o título de cada documento, gerando vetores densos normalizados de dimensão 384.
- Cálculo de Similaridade de Cosseno entre embeddings SBert.
- Limiar de similaridade utilizado para comparações diretas foi fixado em $\theta = 0{,}60$.

Nesta comparação, adotou-se $\theta = 0{,}60$ para SBert apenas para fins de paralelismo com a versão inicial. No entanto, como os vetores SBert incorporam semântica, a cobertura tende a ser muito maior, mesmo com o mesmo limiar.

5.3 TABELAS DE COMPARAÇÃO E MÉTRICA "MATCHING"

Para cada *departamento ou categoria* de documentos (por exemplo, RISK, PUBLIC RELATIONS, HUMAN RESOURCES etc.), calculou-se:

- **Total de Linhas:** número de documentos avaliados naquele conjunto (títulos únicos).
- **Total de Filhos:** número de documentos identificados como "filhos" naquele conjunto.
- **Filhos sem Pai Encontrado (Original / SBert):** quantidade de documentos-filho que *não* encontraram pai em cada versão.
- **Filhos com Pai Encontrado (Original / SBert):** quantidade de documentos-filho que *encontraram* algum documento-pai em cada versão.

- **Total Encontrados em "Original":** número de pares (filho–pai) detectados pela versão inicial.
- **Total Matching:** número de pares (filho–pai) detectados pela versão final (SBert) que coincidiram exatamente com os pares encontrados na versão inicial.
- **% Matching:** porcentagem de pares "originais" que também foram detectados pela versão final, i.e.,

$$\%\,\mathrm{Matching} = \frac{\text{Total Matching}}{\text{Total Encontrados em Original}} \times 100\%.$$

As Tabelas 9 a 20 exemplificam esses valores para alguns grupos de documentos. Para cada grupo, apresentamos *duas subcolunas* indicando "Original" (BOW + Cosseno) e "SBert" (SBert + Cosseno).

Tabela 9 – Comparativo de Cobertura e "Matching" – Categoria I

Label	Original	SBert	Total Matching	% Matching
Total de linhas	35	35	—	—
Total de filhos	31	31	—	—
Filhos sem pai encontrado	30	30	—	—
Filhos com pai encontrado	1	1	1	100 %

Tabela 10 – Comparativo de Cobertura e "Matching" – Categoria II

Label	Original	SBert	Total Matching	% Matching
Total de linhas	134	134	—	—
Total de filhos	118	118	—	—
Filhos sem pai encontrado	60	34	—	—
Filhos com pai encontrado	58	85	55	94,83 %

Tabela 11 – Comparativo de Cobertura e "Matching" – Categoria III

Label	Original	SBert	Total Matching	% Matching
Total de linhas	5	5	—	—
Total de filhos	3	3	—	—
Filhos sem pai encontrado	2	2	—	—
Filhos com pai encontrado	1	1	1	100 %

Tabela 12 – Comparativo de Cobertura e "Matching" – Categoria IV

Label	Original	SBert	Total Matching	% Matching
Total de linhas	105	105	—	—
Total de filhos	99	99	—	—
Filhos sem pai encontrado	53	30	—	—
Filhos com pai encontrado	46	69	36	78,26 %

Tabela 13 – Comparativo de Cobertura e "Matching" – Categoria V

Label	Original	SBert	Total Matching	% Matching
Total de linhas	59	59	—	—
Total de filhos	52	52	—	—
Filhos sem pai encontrado	42	27	—	—
Filhos com pai encontrado	10	25	19	100 %

Tabela 14 – Comparativo de Cobertura e "Matching" – Categoria VI

Label	Original	SBert	Total Matching	% Matching
Total de linhas	224	224	—	—
Total de filhos	216	216	—	—
Filhos sem pai encontrado	144	96	—	—
Filhos com pai encontrado	72	120	70	97,22 %

Tabela 15 – Comparativo de Cobertura e "Matching" – Categoria VII)

Label	Original	SBert	Total Orig.	Total Matching	% Matching
Total de linhas	292	292	—	—	—
Total de filhos	281	281	—	—	—
Filhos sem pai encontrado	185	116	—	—	—
Filhos com pai encontrado	96	165	83	86,46 %	

Tabela 16 – Comparativo de Cobertura e "Matching" – Categoria VIII

Label	Original	SBert	Total Matching	% Matching
Total de linhas	140	140	—	—
Total de filhos	133	133	—	—
Filhos sem pai encontrado	58	33	—	—
Filhos com pai encontrado	75	100	74	98,67 %

Tabela 17 – Comparativo de Cobertura e "Matching" – Categoria IX

Label	Original	SBert	Total Matching	% Matching
Total de linhas	8	8	—	—
Total de filhos	2	2	—	—
Filhos sem pai encontrado	1	1	—	—
Filhos com pai encontrado	1	1	1	100 %

Tabela 18 – Comparativo de Cobertura e "Matching" – Categoria X

Label	Original	SBert	Total Matching	% Matching
Total de linhas	43	43	—	—
Total de filhos	35	35	—	—
Filhos sem pai encontrado	26	21	—	—
Filhos com pai encontrado	9	15	8	88,89 %

Tabela 19 – Comparativo de Cobertura e "Matching" – Categoria XI

Label	Original	SBert	Total Matching	% Matching
Total de linhas	259	259	—	—
Total de filhos	250	259	—	—
Filhos sem pai encontrado	213	204	—	—
Filhos com pai encontrado	37	56	25	67,57 %

Tabela 20 – Comparativo de Cobertura e "Matching" – Categoria XII

Label	Original	SBert	Total Matching	% Matching
Total de linhas	147	147	—	—
Total de filhos	139	139	—	—
Filhos sem pai encontrado	96	91	—	—
Filhos com pai encontrado	43	48	30	69,77 %

5.4 ANÁLISE DOS RESULTADOS

5.4.1 Ganho de Cobertura

Observa-se que, em quase todas as categorias, a versão SBert + Cosseno encontrou *muito mais documentos-filho com pai identificado* mais resultados do que a versão inicial (BOW + Cosseno). Por exemplo:

- Em `Categoria I`, a versão inicial identificou 59 pares (50 % de cobertura), enquanto SBert localizou 85 pares (≈71 % de cobertura).
- Em `Categororia VI`, a cobertura saltou de 72 (33 %) para 120 (55 %).
- Em `Categororia VII`, a versão inicial encontrou 97 pares (≈34 % de cobertura), e SBert encontrou 165 pares (≈59 %).
- Em `Categororia VIII`, subiu de 75 (≈56 %) para 100 (≈75 %).
- Na categoria `Categororia XI`, aumentou de 38 (≈15 %) para 56 (≈22 %).

Em todos os casos, SBert superou a versão inicial, muitas vezes duplicando ou triplicando a quantidade de pais encontrados. Isso confirma que a semântica captada pelos embeddings SBert, mesmo usando apenas títulos, aumenta significativamente o *recall* na detecção de relações pai–filho.

5.4.2 Taxa de "Matching" com Pares Originais

A coluna "% Matching" mostra a proporção de pares detectados pela versão inicial que também foram recuperados pela versão SBert. Em linhas gerais:

- Alta consistência ($\geq 90\%$) nas categorias `I` (100 %), `III` (100 %), `V` (100 %), `VI` (97,22 %), `VIII` (98,67 %).
- Moderada consistência ($75\% \leq \%\,\mathrm{Matching} < 90\%$) em `II` (94,83 %), `VII` (86,46 %), `X` (88,89 %).
- Mais baixo em `XII` (69,77 %), `XI` (67,57 %).

Esses números indicam que, quando a versão inicial (BOW + Cosseno) identificou um par de títulos com cosseno $\geq 0{,}60$, SBert na grande maioria dos casos também capturou o mesmo par (correspondência), confirmando a *precisão* dos pares originais. As poucas discrepâncias (por exemplo, 67,57 % em Engenharia) podem ocorrer quando SBert considera que uma correspondência original ficava abaixo do limiar semântico de 0,60, ou, mais provável, quando existem múltiplos candidatos próximos, alterando o ranking mínimo, já que nesses casos em que o "% Matching" foi mais baixo, o número de documentos era muito alto.

5.4.3 Limitações da Versão Inicial

1. **Baixa Cobertura Absoluta:** A abordagem BOW + Cosseno, com limiar de 60 %, deixou de identificar a maior parte dos pares pai–filho que, semânticamente, existiam. Em algumas categorias, menos de 20 % dos filhos encontraram um pai.
2. **Dependência de Vocabulário Exato:** Documentos cujos títulos possuíam sinônimos ou pequenas variações ("Gestão de Risco" vs. "Gerenciamento de Riscos") não atingiam θ, apesar de terem relação conceitual forte.
3. **Falsos Negativos Elevados:** Na `Categororia VII`, apenas 34,16 % dos filhos encontraram pai; ou seja, mais de 65 % das relações válidas foram perdidas.

Por outro lado, sempre que a similaridade de cosseno entre títulos excedia 0,60, a avaliação subjetiva confirmou consistentemente a existência de relação pai–filho, mostrando que a versão inicial tinha *alta precisão* nos pares reconhecidos, mas alto custo em *recall*.

5.5 EXPERIMENTOS FUTUROS

5.5.1 Proposta de Abordagem Híbrida

Diante da alta precisão observada nos pares originais (BOW + Cosseno) e do elevado recall de SBert, propõe-se, para uso prático em ambiente corporativo ao qual o corpo ou resumo dos documentos não está disponível, uma **pipeline híbrida** sobre os títulos:

1. **Etapa 1 – BOW + Cosseno com Limiar Elevado ($\theta' > 0{,}60$)**:
   - Executar vetorização BOW nos títulos e calcular cosseno.

- Usar um limiar θ' mais conservador (por exemplo, 0,75 ou 0,80) de modo a garantir que apenas pares de títulos quase idênticos (ou que compartilham a maior parte dos termos) sejam automaticamente classificados como pai–filho.
- Essa etapa captura cases de *alta similaridade sintática* com probabilidade quase certa de relação, mantendo a *precisão máxima*.

2. **Etapa 2 – SBert + Cosseno para Restantes**:
   - Para todos os filhos que *não* encontraram pai na Etapa 1, aplicar vetorização SBert nos títulos e calcular similaridade de cosseno.
   - Usar um limiar intermediário ($\theta = 0{,}60$ ou ajustado) para capturar relações temáticas não identificadas pela Etapa 1.
   - Essa etapa amplia significativamente o *recall*, mantendo, em geral, boa precisão, conforme demonstrado neste experimento.
3. **Justificativa**:
   - A *Etapa 1* documenta explicitamente pares de "títulos quase iguais" sem incorrer em custos computacionais elevados.
   - A *Etapa 2* elimina a maior parte dos falsos negativos ao incorporar semântica SBert, mas só é aplicada onde a sintaxe BOW falha, reduzindo o número de comparações SBert (e, portanto, requisitos computacionais).
   - Em ambiente corporativo, esse pipeline híbrido torna-se mais eficiente, pois:
     - Reduz o tempo de inferência geral (apenas uma fração do conjunto passa por SBert).
     - Garante alta precisão nos pares mais óbvios (BOW alto $\rightarrow$ relação quase certa).
     - Aproveita o poder semântico de SBert nos casos em que os títulos diferem lexicalmente mas são semanticamente relacionados.

5.5.2 SBert em Resumos de Documentos

Embora este experimento tenha operado exclusivamente sobre os *títulos*, o **objetivo final** para a empresa é usar SBert em *resumos completos* dos documentos (ou em trechos principais), de modo a:

- **Aumentar ainda mais o Recall**: Resumos contêm frases, contextos e termos específicos que não aparecem nos títulos, permitindo capturar relações pai–filho mais sutis.
- **Reduzir Falsos Positivos**: Em alguns casos, títulos podem ser ambíguos ("Procedimento Operacional" pode referir-se a múltiplos documentos de departamentos diferentes). Avaliar o texto do resumo ajuda a confirmar a relevância contextual.
- **Manter Segurança da Informação**: Para garantir confidencialidade, planeja-se extrair *somente* o "trecho resumo" dos documentos que já são de domínio interno ou aprovados para processamento, evitando expor textos completos.
- **Ajustar Limiar Semântico**: Em resumos, poderá ser necessária calibrar o limiar de cosseno SBert (talvez elevar para 0,65 ou 0,70) para evitar "falsos positivos" gerados por termos genéricos.

O experimento apresentado serve como etapa preparatória para esse cenário: valida-se que SBert, mesmo operando sobre títulos, já supera amplamente BOW, e sugere-se que o ganho sobre resumos completos será ainda maior.

5.6 RESUMO DOS PRINCIPAIS ACHADOS

1. A **versão final (SBert + Cosseno)** alcançou coberturas de pai–filho entre 55 % e 75 % em muitas categorias, comparada a coberturas abaixo de 35 % na versão inicial (BOW + Cosseno).
2. A **precisão** dos pares originalmente identificados (BOW + Cosseno, limiar 0,60) foi confirmada em $\geq 85\,\%$ dos casos, conforme a taxa de "Matching", o que mostra que a versão inicial, embora limitada, produzia pares confiáveis.
3. A adoção de SBert em resumos de documentos deve elevar ainda mais cobertura e qualidade, reduzindo falsos negativos remanescentes.
4. Uma *pipeline híbrida* (BOW conservador + SBert para o restante) equilibra *precisão*, *recall* e *eficiência computacional*, sendo recomendada para aplicação prática imediata.

Com isso, encerramos a análise comparativa entre versões do projeto, evidenciando como a aquisição de embeddings semânticos (SBert) representa salto substancial em performance para a restauração de relações hierárquicas em documentos corporativos.

6 CONCLUSÃO

Neste trabalho, foi abordado o problema da restauração automática das relações hierárquicas de pai–filho entre documentos corporativos (TULLY; AL., 2024), a partir apenas de seus títulos e resumos, nos testes, ou apenas títulos, com dados parciais reais. Inicialmente, implementou-se uma solução baseada em Bag of Words e Similaridade de Cosseno com um limiar de 0,60, que mostrou alta precisão, porém baixa cobertura (*recall*). Em seguida, conduziram-se experimentos comparativos envolvendo cinco técnicas de vetorização (TF-IDF, Word2Vec, FastText, SBert e Embeddings OpenAI) combinadas com três métricas de similaridade (Cosseno, Distância Euclidiana e Jaccard). Para cada combinação, calculou-se a *Similaridade Temática* média sobre os três itens mais similares de uma base de 9 000 artigos científicos, validada por análise subjetiva assistida por GPT-4o.

6.1 PRINCIPAIS ACHADOS

As técnicas baseadas em *embeddings* transformer (SBert e OpenAI) superaram amplamente TF-IDF, Word2Vec e FastText, obtendo médias de Similaridade Temática de aproximadamente 0,86 (SBert+Cosseno/Euclidiana) e 0,85 (OpenAI+Cosseno/Euclidiana).

O modelo SBert apresentou o melhor trade-off *desempenho* $\times$ *custo*, pois roda localmente sem custos de API e mantém Coseno e Euclidiana intercambiáveis (média $=$ 0,8619). A execução local do SBert também se destaca por não oferecer nenhum risco de vazamento de dados ao utilizar APIs externas.

Os Métodos baseados em vetores de palavras (Word2Vec, FastText) se beneficiaram de Jaccard ponderado, alcançando até 0,74 de ST, mas ainda ficaram atrás de SBert.

A comparação direta entre a versão inicial (Bag of Words $+$ Cosseno, limiar $=$ 0,60) e SBert $+$ Cosseno mostrou que SBert encontrou, em média, duas a três vezes mais relações pai–filho, mantendo acima de 85 % de "matching" com os pares originais.

Ao fim dos experimentos, destacou-se como melhor combinação de técnicas, a dupla *SBert $+$ Similaridade de Cosseno* (ou, equivalentemente, Distância Euclidiana), que mostrou-se a mais eficaz, oferecendo alta cobertura ($\geq$ 55 % a 75 % em diferentes domínios), excelente consistência e custo zero de API.

6.2 POSSÍVEIS MELHORIAS E TRABALHOS FUTUROS

Em trabalhos futuros, uma das primeiras frentes de evolução seria implementar um pipeline híbrido que combine a abordagem Bag of Words com Similaridade de Cosseno — utilizando um limiar conservador (por exemplo, 0,75) para capturar automaticamente pares de alta similaridade sintática — seguido de uma segunda etapa com SBert + Cosseno aplicada apenas aos casos restantes. Essa estratégia permitiria manter elevada precisão nos casos óbvios, reduzir o número de inferências SBert e, ao mesmo tempo, aumentar significativamente o recall.

Além disso, é recomendável estender o uso de SBert para além dos títulos, aplicando o modelo sobre resumos ou trechos principais dos documentos. A incorporação de contexto adicional proveniente dos resumos deve reduzir ambiguidades de títulos semelhantes e elevar ainda mais a cobertura de relações pai–filho, sobretudo em ambientes corporativos que exigem segurança e confidencialidade na manipulação de texto completo.

Outra linha de pesquisa consiste em explorar variantes de modelos transformer específicos para o domínio empresarial — seja quantizando SBert, treinando versões fine-tuned em documentos corporativos ou avaliando arquiteturas mais recentes —, bem como investigar métricas híbridas que combinem Cosseno e Jaccard ponderado para refinar as decisões de similaridade. Finalmente, recomenda-se validar a metodologia em conjuntos de dados reais, integrando os resultados a um protótipo ou API interna com interface de revisão manual e coleta de feedback, a fim de calibrar limiares e otimizar o balanceamento entre precisão, recall e eficiência computacional em ambientes de produção com grande volume de documentos.

Em síntese, os experimentos confirmam que embeddings semânticos baseados em transformer representam um salto qualitativo no restauro de hierarquias documentais e pavimentam o caminho para soluções híbridas e escaláveis em ambiente corporativo.

REFERÊNCIAS

AMAN. *Word Embedding Techniques: From TF-IDF to Contextual Embeddings*. 2023. Aman's AI Journal. Disponível em: <https://aman.ai/primers/ai/word-vectors/>.

ASUDANI, D. S.; NAGWANI, N. K.; SINGH, P. Impact of word embedding models on text analytics: A review. *Artificial Intelligence Review*, Springer, v. 56, p. 10345–10425, 2023. Disponível em: <https://link.springer.com/article/10.1007/s10462-023-10419-1>.

BAYTAS, C.; RUEDIGER, D. *Generative AI in Higher Education: The Product Landscape*. 2024. Ithaka S+R Issue Brief. Disponível em: <https://sr.ithaka.org/publications/generative-ai-in-higher-education/>. Acesso em: 15 jun. 2025.

Kaggle user arashnic. *Urban Sound Dataset*. 2025. Kaggle Datasets. Disponível em: <https://www.kaggle.com/datasets/arashnic/urban-sound>. Acesso em: 15 jun. 2025.

OpenAI. *New and improved embedding model (text-embedding-ada-002)*. 2022. OpenAI Official Blog. Disponível em: <https://openai.com/blog/new-and-improved-embedding-model>. Acesso em: 15 jun. 2025.

REIMERS, N.; GUREVYCH, I. Sentence-bert: Sentence embeddings using siamese bert-networks. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, 2019. p. 3973–3983. Disponível em: <https://ntrs.nasa.gov/api/citations/20220008275/downloads/20220008275_idetc-2022_v2.pdf>.

SIMPLR. *Comparing Popular Embedding Models: Choosing the Right One for Your Use Case*. 2023. DEV Community. Disponível em: <https://dev.to/simplr_sh/comparing-popular-embedding-models-choosing-the-right-one-for-your-use-case-43p1>.

SINGH, A. *Ultimate Guide to Text Similarity With Python*. 2024. NewsCatcher Blog. Disponível em: <https://www.newscatcherapi.com/blog/ultimate-guide-to-text-similarity-with-python>. Acesso em: 15 jun. 2025.

TULLY, T.; AL. et. *2024: The State of Generative AI in the Enterprise*. 2024. Menlo Ventures Report. Disponível em: <https://menlovc.com/2024-the-state-of-generative-ai-in-the-enterprise/>. Acesso em: 15 jun. 2025.

VIDAL, F. *Similarity Distances for Natural Language Processing*. 2021. Medium. Disponível em: <https://flavien-vidal.medium.com/similarity-distances-for-natural-language-processing-16f63cd5ba55>. Acesso em: 15 jun. 2025.

WANG, Z.; AL. et. Large language models are zero-shot text classifiers. *arXiv preprint arXiv:2312.01044*, 2023. Disponível em: <https://arxiv.org/abs/2312.01044>. Acesso em: 15 jun. 2025.