UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

MARTÍN ÁVILA BUITRÓN

IDENTIFICAÇÃO DE TEMAS EMERGENTES EM NOTÍCIAS ATRAVÉS DE MÉTODOS NÃO-SUPERVISIONADOS

TOLEDO

2025

MARTÍN ÁVILA BUITRÓN

IDENTIFICAÇÃO DE TEMAS EMERGENTES EM NOTÍCIAS ATRAVÉS DE MÉTODOS NÃO-SUPERVISIONADOS

Detection of Emerging Topics in News through Unsupervised Learning Models

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia da Computação do Curso de Bacharelado em Engenharia de Computação da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Gustavo Henrique Paetzold

TOLEDO

2025

MARTÍN ÁVILA BUITRÓN

IDENTIFICAÇÃO DE TEMAS EMERGENTES EM NOTÍCIAS ATRAVÉS DE MÉTODOS NÃO-SUPERVISIONADOS

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia da Computação do Curso de Bacharelado em Engenharia de Computação da Universidade Tecnológica Federal do Paraná.

Data de aprovação: 04/dezembro/2025

Prof. Dr. Gustavo Henrique Paetzold
Orientador - Universidade Tecnológica Federal do Paraná

Prof. Dr. Daniel Cavalcanti Jeronymo
Universidade Tecnológica Federal do Paraná

Prof. Dr. Álvaro Ricieri Castro e Souza
Universidade Tecnológica Federal do Paraná

Prof. Dr. Leandro Augusto Ensina
Universidade Tecnológica Federal do Paraná

TOLEDO

2025

Dedico este trabalho à minha família, em especial ao meu avô, pelo exemplo e por sempre confiar em mim.

AGRADECIMENTOS

Agradeço a todos que, de alguma forma, contribuíram para esta etapa da minha vida.

Ao Prof. Dr. Gustavo Henrique Paetzold, pela orientação atenta e segura;

Aos colegas de sala, pela parceria; e à Secretaria do Curso, pela constante cooperação;

Especialmente, o meu reconhecimento à minha família, cujo apoio tornou esta jornada possível.

Enfim, A cada pessoa que me acompanhou e incentivou, deixo minha sincera gratidão.

RESUMO

A *Novelty Detection*, ou detecção de novidade em documentos, é uma tarefa desafiadora e de grande relevância na atualidade. A literatura trata este problema sob distintas abordagens, utilizando aprendizado supervisionado, não supervisionado, autosupervisionado, entre outros. Este tema é especialmente significativo no campo do Processamento de Linguagem Natural, visto que o intuito é diferenciar textos que pertencem a um conjunto já conhecido daqueles que trazem informações inéditas ou emergentes. Neste trabalho, propõe-se investigar distintos métodos não supervisionados de detecção de novidade em um conjunto de dados de notícias, com o objetivo de compará-los com métodos supervisionados. Foram avaliados métodos como *Local Outlier Factor*, *Isolation Forest* e *Elliptic Envelope*, e uma abordagem moderna baseada na arquitetura RAG com modelos de linguagem de grande escala, comparando-os com *baselines* estabelecidos na literatura. As métricas utilizadas incluem precisão, *recall*, F1-*score* e acurácia. Os resultados demonstraram que o método LOF alcançou desempenho promissor, com F1 de 80,90% e acurácia de 85,80%, em comparação com os *baselines* do estado da arte.

Palavras-chave: detecção de anomalias; outliers ; classificadores; llm.

ABSTRACT

Novelty Detection in documents is a challenging task of great relevance today. The literature addresses this problem through different approaches, using supervised learning, unsupervised learning, self-supervised learning, among others. This topic is especially significant in the field of Natural Language Processing, as the goal is to differentiate texts belonging to a known set from those bringing new or emerging information. In this work, we propose to investigate different unsupervised models for novelty detection in a news dataset, with the objective of comparing them with supervised models. Models such as *Local Outlier Factor*, *Isolation Forest*, and *Elliptic Envelope* were evaluated, along with a modern approach based on the RAG architecture with Large Language Models, comparing them with baselines established in the literature. The metrics used include precision, *recall*, F_1*-score*, and accuracy. The results demonstrated that the Local Outlier Factor model achieved promising performance, with an F1 of 80.90% and accuracy of 85.80%, compared to state-of-the-art baselines.

Keywords: anomaly detection; outliers ; classifiers; llm; .

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Diariamente, milhares de artigos são publicados em portais e jornais, tornando praticamente inviável para o leitor comum identificar, de imediato, quais textos realmente introduzem informação nova. Diante desse cenário, destaca-se a detecção de novidade (*novelty detection*). Dentro desse campo, na literatura, o conceito associa-se ao ciclo de *Knowledge Discovery in Databases* (KDD), definido como "o processo não trivial de identificar informações válidas, novas, potencialmente úteis e, em última análise, conhecimento compreensível a partir dos dados" (BREUNIG *et al.*, 2000). Nesse enquadramento, a detecção de novidade, busca desenvolver sistemas capazes de identificar padrões inéditos e informações exclusivas em comparação com um conjunto de referências já estabelecido.

O entendimento desses sistemas ocorre por meio da diferenciação entre anomalias, *outliers* e ruído. Na literatura clássica, Hawkins (1980) descreve um *outlier* como "uma observação que se desvia tanto das demais que levanta suspeita de ter sido gerada por um mecanismo distinto". Aggarwal (2017) denomina anomalias como instâncias que não se ajustam ao comportamento normal definido. Assim, enquanto *outlier* enfatiza o desvio estatístico, anomalia é mais abrangente, aplicado a qualquer instância que se comporte de maneira inesperada. Por sua vez, Chandola, Banerjee e Kumar (2009) definem ruído como um fenômeno presente nos dados que não interessa ao analista e, simultaneamente, dificulta a análise. Dessa forma, o ruído representa dados indesejados que devem ser interpretados e removidos, ao passo que as anomalias ou *outliers* podem conter informações relevantes. Teng, Chen e Lu (1990) reforçam a ideia que detectar anomalias e remover ruído são atividades relacionadas, porém distintas. Diante dessa distinção, o grande desafio da detecção de novidade consiste em discernir o ruído e, simultaneamente, identificar *outliers* que introduzem informação nova no conjunto de notícias, diferenciando-os de anomalias que podem não necessariamente representar novidades para o sistema.

Sob o ponto de vista metodológico, a detecção de novidade então pode ser entendida como uma *generalização* das técnicas de classificação para cenários em que apenas a classe "normal" está disponível. Aggarwal (2017) denomina essa adaptação como *one-class analogs*. Trata-se de algoritmos supervisionados tradicionais, como *Support Vector Machines* (SVM) ou árvores de decisão, que são modificados para modelar apenas o comportamento da classe normal. Essa modelagem ocorre, geralmente, por meio de um hiperplano que engloba todo o conjunto de dados normais. A partir disso, surgem variantes como a *One-Class SVM* ou o *Isolation Forest*, projetadas especificamente para esse tipo de tarefa.

Neste contexto, o presente trabalho é voltado ao estudo dos principais desafios enfrentados pelos sistemas de detecção de novidade em documentos de notícia. Propõe-se analisar sistematicamente diferentes métodos não supervisionados e abordagens contemporâneas baseadas em modelos de linguagem de grande escala, com o intuito de identificar quais técnicas apresentam melhor desempenho comparado com o estado da arte. Dado o caráter **não su-**

pervisionado da abordagem adotada, o treinamento e teste são realizados evento por evento, respeitando as particularidades lexicais e semânticas inerentes a cada categoria temática. Essa estratégia não apenas preserva a especificidade linguística de domínios distintos (como política, esportes ou negócios), mas também se aproxima de cenários reais de aplicação. Desta maneira, os resultados obtidos visam oferecer diretrizes claras para futuras aplicações nessa área.

1.1 Objetivos

1.1.1 Objetivo geral

O objetivo geral deste trabalho é investigar a eficácia de métodos não supervisionados para a detecção de novidade em documentos de notícia, fornecendo diretrizes claras quanto à eficiência e aplicabilidade dessas técnicas em contextos jornalísticos.

1.1.2 Objetivos específicos

- Realizar levantamento bibliográfico sobre métodos e algoritmos não supervisionados empregados na detecção de novidade;
- Identificar e analisar conjuntos de dados relevantes para experimentos em detecção de novidade;
- Desenvolver e avaliar técnicas não supervisionadas e baseadas em arquitetura RAG através de protocolos experimentais bem definidos;
- Comparar o desempenho das técnicas propostas com *baselines* estabelecidos na literatura, utilizando o conjunto de dados TAP-DLND 1.0 como referência.

1.2 Justificativa

A relevância deste estudo está diretamente ligada à sobrecarga de informações no jornalismo atual. No contexto brasileiro, grandes portais de notícias como G1, UOL e Folha de S.Paulo publicam centenas de matérias diariamente. Com esse volume massivo de reportagens publicadas por esses portais, torna-se desafiador para jornalistas, pesquisadores e leitores identificar rapidamente o que é realmente novo e o que representa apenas uma repetição. Ferramentas automáticas de detecção de novidade surgem como uma solução promissora, mas grande parte dos avanços recentes depende de recursos específicos, muitas vezes limitados a conteúdos em inglês.

Além disso, embora modelos de linguagem de larga escala tenham se destacado em tarefas como tradução e resumo, a literatura demonstra que métodos não supervisionados con-

tinuam competitivos, especialmente em cenários sem dados rotulados (NAIR, 2023). A utilização de um banco de dados jornalístico de *benchmark* atende à necessidade de um ponto de partida sólido para a pesquisa.

Diante desse contexto, este trabalho investiga a aplicação de diferentes métodos não supervisionados para detecção de novidade em documentos textuais. O objetivo principal é avaliar e comparar o desempenho de técnicas como *LOF*, *Isolation Forest* e *Elliptic Envelope*, além de abordagens modernas baseadas em arquitetura RAG com LLMs, utilizando um conjunto de dados de notícias. Busca-se estabelecer uma análise comparativa entre métodos não supervisionados e supervisionados, contribuindo para a compreensão de suas potencialidades e limitações nesta tarefa. Dessa forma, contribui para a comunidade acadêmica frente à necessidade de diferentes soluções em um contexto onde a detecção de novidade se torna cada vez mais essencial.

1.3 Estrutura do documento

Este documento está organizado em cinco capítulos. O **Capítulo 1** apresenta a contextualização da detecção de novidade em documentos, além dos objetivos da pesquisa. O **Capítulo 2** revisa os fundamentos teóricos, abordando o conceito de corpus, técnicas de representação vetorial, aprendizado de máquina aplicado à detecção de novidade, métodos de detecção e métricas associadas, além de exemplos e discussão sobre trabalhos relacionados. O **Capítulo 3** detalha a metodologia proposta, explicitando as etapas e procedimentos adotados. O **Capítulo 4** expõe os resultados esperados e as métricas de avaliação que utilizadas para avaliar a eficiência dos métodos investigados. Finalmente, o **Capítulo 5** apresenta as conclusões da pesquisa.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta os fundamentos teóricos e metodológicos que sustentam o desenvolvimento deste trabalho. Inicialmente, são abordados os conceitos essenciais de Processamento de Linguagem Natural (PLN). Em seguida, explora-se a detecção de novidade, discutindo a terminologia adotada, a taxonomia dos métodos existentes e uma revisão dos algoritmos clássicos. Posteriormente, são apresentados os avanços recentes em PLN proporcionados pela arquitetura Transformer, com destaque para a técnica de Geração Aumentada por Recuperação (RAG). Por fim, são descritas as principais métricas de avaliação utilizadas em tarefas de classificação binária.

2.1 Conceitos e fundamentos de processamento de linguagem natural

Para entender os diferentes tipos de processamento de linguagem, é necessário primeiro compreender em qual categoria de aprendizado de máquina essa tarefa se encaixa. Como define Samuel (1959) , “aprendizado de máquina é o campo de estudo que dá aos computadores a habilidade de aprender sem serem explicitamente programados”. Esse campo é um subdomínio da inteligência artificial voltado ao desenvolvimento de modelos baseados em dados, e pode ser dividido em diferentes abordagens.

2.1.1 Aprendizado supervisionado vs aprendizado não supervisionado

O aprendizado supervisionado, sendo o primeiro tipo, em que “os algoritmos recebem dados rotulados e tentam aprender uma função que relacione entradas às saídas desejadas” (GéRON, 2021). Essa técnica é útil quando se dispõe de grandes volumes de dados anotados, permitindo treinar classificadores a partir de exemplos previamente conhecidos. Por outro lado, no aprendizado não supervisionado “não há rótulos, e o objetivo é encontrar estrutura nos dados” (BISHOP, 2006). Essa abordagem é particularmente indicada em tarefas onde as anomalias são raras, desconhecidas ou sem rótulos.

No caso específico do processamento de linguagem para detecção de novidade, os métodos empregados podem variar entre supervisionados ou não supervisionados. Isso se deve à natureza intrínseca do conjunto de dados utilizado. Em todo caso, este trabalho observa o problema de detecção de novidades como um problema de detecção de *outliers*. Esta atribuição terminológica será discutida na Seção 2.2.1. Essas definições mostram que, mesmo quando se utilizam algoritmos que normalmente são aplicados em tarefas supervisionadas, os modelos podem operar de forma não supervisionada. Neste caso, o objetivo é determinar se uma nova instância se encaixa no padrão aprendido.

2.1.2 *Corpus* e documentos no pré-processamento linguístico

No pré-processamento linguístico, subárea da recuperação de informação, surge o conceito *corpus*, proveniente do latim, alusão a corpo. Manning, Raghavan e Schütze (2008), no livro *Introduction to Information Retrieval*, definem um *corpus* como uma coleção organizada de documentos, cuja finalidade principal é recuperar ou abstrair informações relevantes. Por sua vez, McEnery e Hardie (2012), no livro *Corpus Linguistics*, detalham que um *corpus* deve ser composto por textos preservados integralmente, acompanhados por metadados contextuais (como autor, data, fonte e gênero), codificados e segmentados sistematicamente (por exemplo, por título, parágrafo e sentença). Com base nessas definições, os documentos utilizados nesta pesquisa correspondem a um *corpus* jornalístico, estruturado conforme as caraterísticas descritas acima.

Para trabalhar neste tipo de documentos, é necessário executar uma etapa inicial de pré-processamento para extrair informações úteis a partir do texto bruto. Segundo Grus (2019), o processamento de texto ou Processamento de Linguagem Natural (PLN) envolve técnicas computacionais que permitem a manipulação e análise automática da linguagem humana. Como é possível observar na Tabela 1, essas técnicas compreendem etapas específicas como **limpeza**, ***tokenização***, **remoção de *stopwords*** e **lematização**. O mesmo autor destaca que a limpeza dos dados é um passo essencial, pois frequentemente há muita informação irrelevante que precisa ser removida para evitar erros de interpretação. Por exemplo, notícias extraídas de sites frequentemente incluem publicidade, *tags HTML* ou outros elementos.

Tabela 1 – Exemplo de pré-processamento linguístico em etapas

Etapa	**Transformação da Frase**
Texto original	`<section> O presidente falou hoje no evento. Clique aqui para saber mais!`
Limpeza	`O presidente falou hoje no evento. Clique aqui para saber mais`
Tokenização	`["O", "presidente", "falou", "hoje", "no", "evento", "Clique", "aqui", "para", "saber", "mais"]`
Remoção de *stopwords*	`["presidente", "falou", "evento", "Clique", "saber"]`
Lematização	`["presidente", "falar", "evento", "clicar", "saber"]`

Fonte: Autoria própria (2025).

A *tokenização*, por sua vez, refere-se ao processo de dividir o texto em unidades menores chamadas *tokens*, geralmente palavras ou termos individuais (JURAFSKY; MARTIN, 2020). Esse procedimento facilita análises subsequentes, pois possibilita a quantificação e manipulação individualizada dos termos.

Por último, temos os processos de remoção de ***stopwords*** e lematização. A remoção de *stopwords* consiste em eliminar palavras muito comuns e pouco informativas (por exemplo,

artigos, preposições e conjunções), que geralmente não contribuem para a diferenciação do conteúdo textual (MANNING; RAGHAVAN; SCHüTZE, 2008). Já a lematização é o processo de reduzir palavras flexionadas ou derivadas às suas formas base, facilitando o agrupamento de termos semanticamente similares (JURAFSKY; MARTIN, 2020).

2.1.3 Representações Vetoriais

O pré-processamento é uma etapa fundamental no trabalho com textos. Para que algoritmos computacionais possam manipulá-los, é necessário convertê-los em representações numéricas vetoriais. Essas representações variam em complexidade e finalidade. Algumas capturam apenas características léxicas e estatísticas, como a frequência de termos, enquanto outras incorporam informações semânticas, sintáticas e contextuais (GRUS, 2019). Em particular, a **novidade léxica** ocorre quando um texto utiliza combinações ou palavras diferentes, mas sem introduzir novos fatos ou conceitos. Já a **novidade semântica** representa um conceito inédito em relação ao conjunto de referência. A Tabela 2 mostra a diferença entre uma frase léxica e uma frase semântica. Observa-se que, no primeiro caso, há apenas uma reformulação lexical, enquanto no segundo caso surge um elemento informacional novo (a contratação de um novo treinador), caracterizando de fato uma novidade semântica.

Tabela 2 – Exemplo de diferença entre novidade léxica e novidade semântica.

Novidade léxica
Notícia base: “O time venceu a partida por dois gols de diferença.” Nova notícia: “A equipe triunfou no jogo com uma vantagem de dois gols.”
Novidade semântica
Notícia base: “O time venceu a partida por dois gols de diferença.” Nova notícia: “O time anunciou a contratação de um novo treinador após a vitória.”

Fonte: Autoria própria (2025).

Na detecção de anomalias, esses vetores suportam diferentes abordagens. Existem métodos baseados em instâncias que comparam distâncias entre vetores de documentos individuais. Já os métodos de generalização explícita modelam globalmente os dados normais por meio de estruturas compactas, como subespaços ou hiperplanos (AGGARWAL, 2017). Nas próximas seções, serão aprofundados esses métodos, discutindo-se sua relevância no aprendizado não supervisionado, além da exploração das técnicas estatísticas associadas.

Entre os métodos de representação vetorial léxica mais tradicionais, destacam-se o modelo *Bag-of-Words* (BoW) e o *TF-IDF* (*Term Frequency–Inverse Document Frequency*). Conforme apresentado por Grus (2019), a ideia central do TF-IDF é ponderar os termos com base em sua frequência no documento e na raridade nos demais documentos do *corpus*. Isso permite reduzir a influência de palavras muito comuns (por exemplo “o”, “de”, “e”) e valorizar termos mais característicos de cada texto.

As Tabelas 3 e 4 ilustram, de forma simplificada, as diferenças entre as representações BoW e TF-IDF. Na Tabela 3, o BoW trata cada palavra como uma variável independente, considerando apenas o número de ocorrências em cada documento, sem distinguir sua importância relativa. Já o TF-IDF incorpora uma dimensão adicional estatística, onde valores mais altos indicam termos raros e distintivos do documento, enquanto valores próximos de zero correspondem a palavras frequentes e pouco informativas. Assim, quanto mais raro for um termo dentro do *corpus*, maior será o seu peso TF-IDF, refletindo seu potencial de contribuir para a diferenciação entre documentos.

Tabela 3 – Exemplo de representação BoW

Doc	o	gato	dorme	cachorro	late	e
D1	1	1	1	0	0	0
D2	1	0	0	1	1	0
D3	0	1	0	1	0	1

Fonte: Autoria própria (2025).

Tabela 4 – Exemplo de representação TF-IDF

Doc	o	gato	dorme	cachorro	late	e
D1	0.176	0.176	0.477	0	0	0
D2	0.176	0	0	0.176	0.477	0
D3	0	0.176	0	0.176	0	0.477

Fonte: Autoria própria (2025).

Após representar de forma vetorial os documentos, em muitos casos, devido a uma grande quantidade de palavras únicas, surge o problema de uma alta dimensionalidade dos dados, em que os dados se tornam esparsos e as métricas tradicionais de distância perdem eficácia (VERLEYSEN; FRANÇOIS, 2005). Para lidar com este problema, surgiram técnicas que, além de auxiliar na representação vetorial dos documentos, também contribuem para mitigar os efeitos causados por *outliers*. Entre essas técnicas, destaca-se a *Análise de Componentes Principais* (PCA), que transforma os dados para um novo sistema de coordenadas, reduzindo redundâncias e capturando os padrões gerais presentes nos dados (AGGARWAL, 2017). Essa transformação melhora a interpretação das amostras representadas como vetores. Outra técnica relevante é o *Latent Semantic Indexing* (LSI), que como o próprio nome indica, permite capturar relações semânticas latentes entre palavras e documentos.

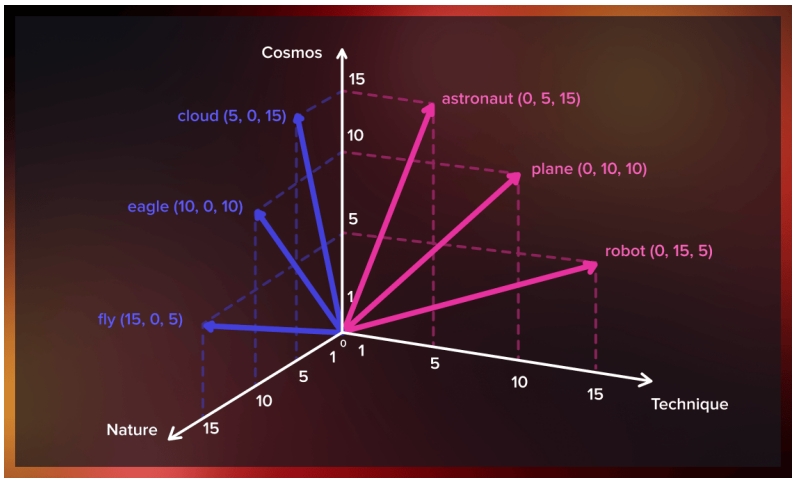
Como mencionado no Capítulo 1, textos geralmente contêm uma grande quantidade de ruído. Nesse contexto, o LSI demonstrou ser eficaz ao gerar representações mais robustas, especialmente por reduzir os efeitos da sinonímia, quando diferentes palavras expressam o mesmo conceito, e da polissemia, quando uma mesma palavra possui múltiplos significados. Como destacado por Aggarwal (2017), tais técnicas melhoram significativamente a qualidade da recuperação de informações em tarefas de similaridade textual.

Um dos avanços mais significativos na área é a capacidade de gerar *embeddings contextuais*. “Um *embedding* é um vetor denso de valores de ponto flutuante (o comprimento do

vetor é um parâmetro que você especifica)" (JURAFSKY; MARTIN, 2020). Ao contrário das representações tradicionais, como *Bag-of-Words* ou *TF-IDF*, que são esparsas e desconsideram a ordem e o contexto, os *embeddings* são vetores que representam palavras, sentenças ou documentos como vetores densos em um espaço multidimensional a partir dos dados. Essa definição ressalta que os *embeddings* são vetores densos em que palavras semanticamente similares são mapeadas para pontos próximos no espaço vetorial (JURAFSKY; MARTIN, 2020).

À vista disso, surgiram modelos de representação densa baseados em vetores contínuos, como o *Word2Vec* (MIKOLOV *et al.*, 2013) e o *Doc2Vec* (LE; MIKOLOV, 2014). Como pode ser observado na Figura 1, esses modelos aprendem representações contextuais, ou seja, capturam o significado das palavras levando em conta o contexto em que estão inseridas. Essa abordagem foi posteriormente superada pelos modelos baseados na arquitetura *Transformer*, proposta por Vaswani *et al.* (2023), que revolucionaram o Processamento de Linguagem Natural. Uma descrição mais aprofundada sobre a arquitetura *Transformer* e seus impactos será apresentada na Seção 2.3.

Figura 1 – Exemplo de Espaço Vetorial Semântico Tridimensional Gerado pelo Word2Vec



Fonte: Logunova (2023).

Finalmente, uma comparação resumida entre dois tipos de representações vetoriais pode ser observado na Tabela 5. As representações léxicas não capturam semântica nem contexto, em comparação com os *embeddings* que oferecem vetores menores, contextuais e semanticamente informativos. Essa diferença é fundamental na escolha da representação para detecção de novidade.

Tabela 5 – Comparação entre representações léxicas e *embeddings*

Característica	**Representações Léxicas (Esparsas)**	***Embeddings* (Densas)**
Tamanho do vetor	Muito grande	Pequeno
Esparsidade	Disperso (zeros)	Denso (todos os valores $\neq 0$)
Captura semântica	Não	Sim
Dependência de contexto	Não	Sim (em *embeddings* contextuais)
Exemplos	Bag-of-Words, TF-IDF	Word2Vec, Doc2Vec, BERT, SBERT

Fonte: Autoria própria (2025).

2.2 Detecção de anomalias e de novidade

2.2.1 Terminologia

Como discutido, a detecção de novidade tem o objetivo de identificar documentos que trazem informações inéditas em relação a um conjunto de referência, normalmente não rotulado (AHMED; MAHMOOD; HU, 2016). Essa abordagem se relaciona com a detecção de anomalias, mas difere na natureza do desvio. A abordagem diferencia-se da detecção de anomalias pela própria definição das anomalias, que representam quaisquer instâncias que não se ajustam ao comportamento normal definido (AGGARWAL, 2017). Enquanto a detecção de anomalias procura qualquer ocorrência fora do padrão, a detecção de novidade concentra-se apenas em conteúdo sem precedentes. Segundo Aggarwal (2017), a detecção de novidades pode ser abordada sob duas perspectivas de aprendizado distintas:

> "*Novelty detection* é uma área estreitamente relacionada à análise de *outliers*, frequentemente estudada em modelos supervisionados, nos quais novas classes de um fluxo de dados são detectadas em tempo real. Contudo, também é investigada em contextos não supervisionados, especialmente nas tarefas de *first story detection* em fluxos de texto." (AGGARWAL, 2017)

Partindo dos conceitos fundamentais, a detecção de novidade pode ser interpretada como um problema não supervisionado de detecção de *outliers*. Por esse motivo, torna-se essencial diferenciá-la do ruído. Ruído inclui variações lexicais irrelevantes, erros de digitação e mudanças estilísticas que não afetam o significado principal do texto (GARCÍA; HERRERA, 2009; AGGARWAL, 2017). Em contrapartida, a detecção de novidade envolve novidade semântica, caracterizada por uma alteração significativa no conteúdo, capaz de introduzir novos elementos ao leitor.

Definir claramente essa distinção entre novidade, anomalia, *outlier* e ruído é responsabilidade do pesquisador e varia conforme os objetivos do estudo. Como destacado por Aggarwal (2017), essa separação é determinante para o desenvolvimento de sistemas robustos de detecção de anomalias.

2.2.2 Taxonomia de métodos

Conforme a taxonomia proposta por Aggarwal (2017), os métodos de detecção de *outliers* podem ser classificados em quatro grandes categorias. A seguir, foi descrita cada uma dessas abordagens, destacando seus princípios gerais e limitações:

- **Métodos estatísticos globais.** Os dados são modelados sob a forma de uma distribuição de probabilidade em forma fechada. Técnicas como o *z-score* ou o teste de Grubbs detectam *outliers* que se desviam significativamente da média. Apesar de simples e eficientes em contextos numéricos, têm aplicação limitada em dados textuais de alta dimensionalidade (AGGARWAL, 2017).

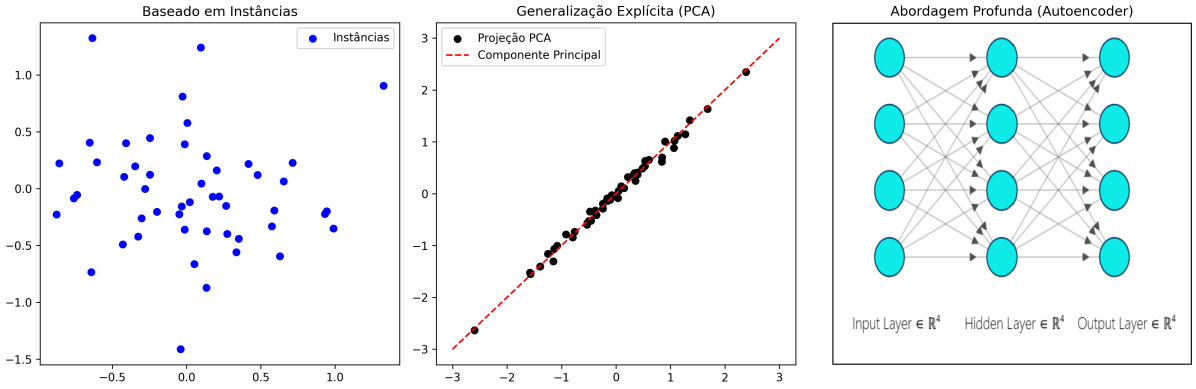
- **Métodos baseados em instâncias.** Representam cada documento como um vetor (por exemplo, usando TF-IDF ou *embeddings*) e avaliam a posição relativa de um documento em relação aos seus vizinhos no espaço vetorial. Algoritmos clássicos incluem a distância k-*NN* e o *Local Outlier Factor* (LOF) (BREUNIG *et al.*, 2000). São métodos tradicionalmente utilizados, mas sensíveis à presença de ruído e à escolha de hiperparâmetros.

- **Métodos de generalização explícita.** Tentam modelar o comportamento normal dos dados de forma compacta, usando estruturas como hiperplanos (*One-Class SVM*) ou subespaços (PCA). A anomalia é estimada com base na distância de um ponto em relação a essa estrutura global. Tais modelos são eficazes quando os dados normais seguem padrões bem definidos (AGGARWAL, 2017).

- **Abordagens profundas.** Incluem redes neurais, como os *autoencoders*, treinados para reconstruir apenas exemplos normais com baixa perda. A anomalia é detectada quando o erro de reconstrução ultrapassa um determinado limiar. Conforme apontam Chalapathy e Chawla (2019) embora esses modelos ofereçam boa capacidade de modelagem não linear, eles exigem maior volume de dados e poder computacional, o que pode limitar sua aplicação em cenários exploratórios.

Como os rótulos (*labels*) de *outliers* raramente estão disponíveis em *corpus* jornalísticos, a maioria dos estudos nessa área recorre a técnicas de aprendizado não supervisionado (AGGARWAL, 2017). Finalmente, a Figura 2 apresenta um exemplo de representação visual das principais abordagens adotadas, usando dados sintéticos, resumindo as abordagens explicadas anteriormente.

Figura 2 – Exemplo de comparação entre diferentes abordagens



Fonte: Autoria própria (2025).

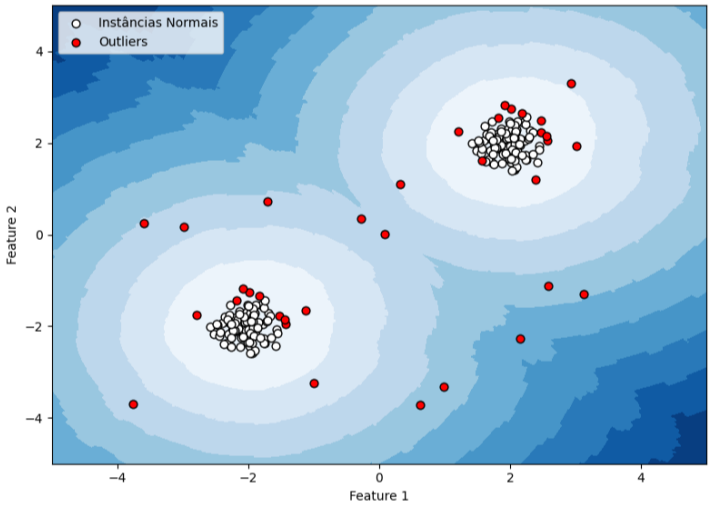
2.2.3 *Local Outlier Factor* (LOF)

Para compreender o algoritmo *LOF*, é importante destacar que ele é um método projetado para identificar ***outliers*** em espaços multidimensionais. Essa característica permite sua aplicação tanto em *Outlier Detection* quanto em *Novelty Detection*.

> O LOF introduz um fator de *outlier* local para cada objeto no conjunto de dados, indicando seu grau de discrepância. Este é, até onde sabemos, o primeiro conceito de *outlier* que também quantifica o quão longe um objeto está. O fator de *outlier* é local no sentido de que apenas uma vizinhança restrita de cada objeto é levada em conta (BREUNIG *et al.*, 2000).

Em resumo, o LOF identifica pontos que se desviam significativamente dos demais ao comparar a densidade local de um ponto com a densidade de seus vizinhos mais próximos. A Figura 3, gerado a partir de dados sintéticos, mostra o mapa de densidade gerado pelo algoritmo LOF. As regiões em azul claro indicam áreas de maior densidade, enquanto o azul escuro representa regiões menos densas. O exemplo mostra como pontos localizados em áreas de baixa densidade, especialmente se comparados à densidade de seus vizinhos, tendem a ser classificados como *outliers*.

Figura 3 – Exemplo de *Local Outlier Factor* (LOF)



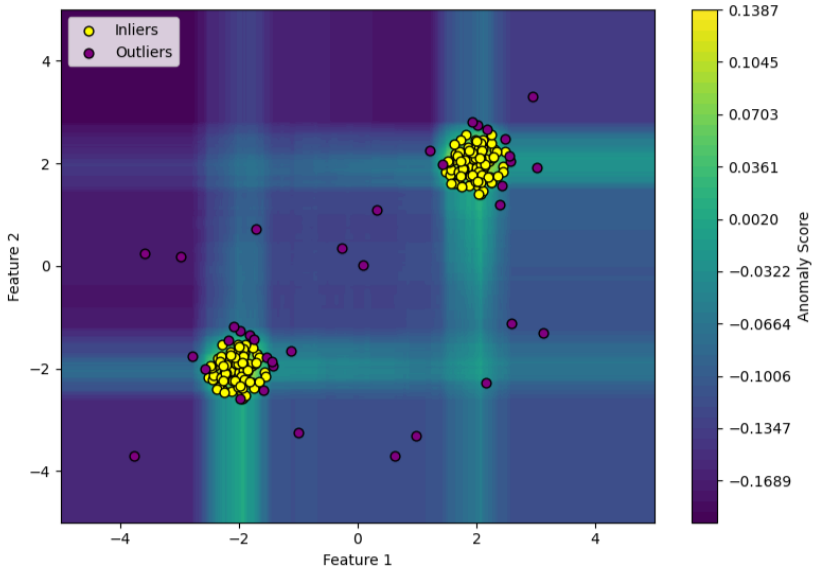
Fonte: Autoria própria (2025).

2.2.4 *Isolation Forest*

Partindo do princípio de que os dados podem ser representados como instâncias passíveis de isolamento, o método *Isolation Forest* surge como uma abordagem distinta a ser testada. Esse algoritmo detecta anomalias utilizando árvores binárias, com complexidade temporal linear e baixa exigência de memória, o que o torna eficiente para lidar com grandes volumes de dados (LIU; TING; ZHOU, 2008). Segundo Liu, Ting e Zhou (2008), a técnica se diferencia por não depender de medidas de distância ou densidade, baseando-se exclusivamente no conceito de isolamento para identificar comportamentos anômalos. Isso significa que o algoritmo busca separar diretamente os pontos fora do padrão em relação ao restante das amostras.

A Figura 4 apresenta o resultado do algoritmo *Isolation Forest* aplicado a dados sintéticos bidimensionais. Os eixos X e Y representam duas características geradas artificialmente. As cores de fundo indicam o grau de anomalia, onde regiões mais escuras representam áreas com maior probabilidade de conter anomalias e regiões claras indicam zonas de dados normais. Os pontos amarelos correspondem a instâncias normais e os pontos roxos representam *outliers* detectados.

Figura 4 – Exemplo de Isolation Forest

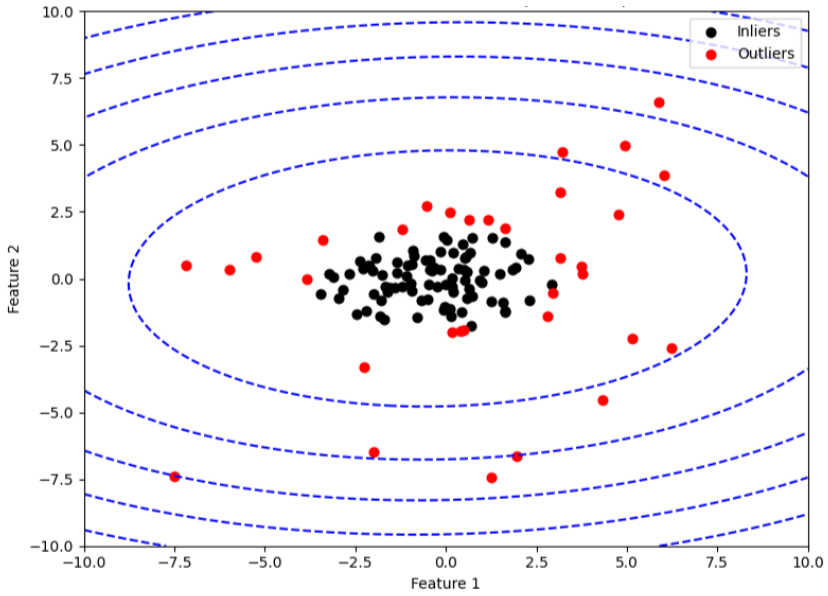


Fonte: Autoria própria (2025).

2.2.5 *Elliptic Envelope*

A partir da definição, “O algoritmo *Elliptic Envelope* é um método de aprendizado de máquina não supervisionado que utiliza estimativas de covariância em dados com distribuição gaussiana.” (Scikit-Learn Developers, 2024). Como pode ser observado no exemplo na Figura 5, o envelope elíptico tenta formar um agrupamento elíptico e se ajusta às principais instâncias dessa classe. Os eixos X e Y representam duas características geradas artificialmente. As elipses azuis indicam os limites estimados do conjunto normal. Os pontos pretos representam instâncias normais, enquanto os pontos vermelhos correspondem a *outliers* detectados. No contexto deste trabalho, as instâncias que estão distantes do agrupamento definido pela elipse do modelo são classificadas como novidades. Esse método é particularmente útil em cenários onde os dados apresentam distribuições aproximadamente normais.

Figura 5 – Exemplo de *Elliptic Envelope*



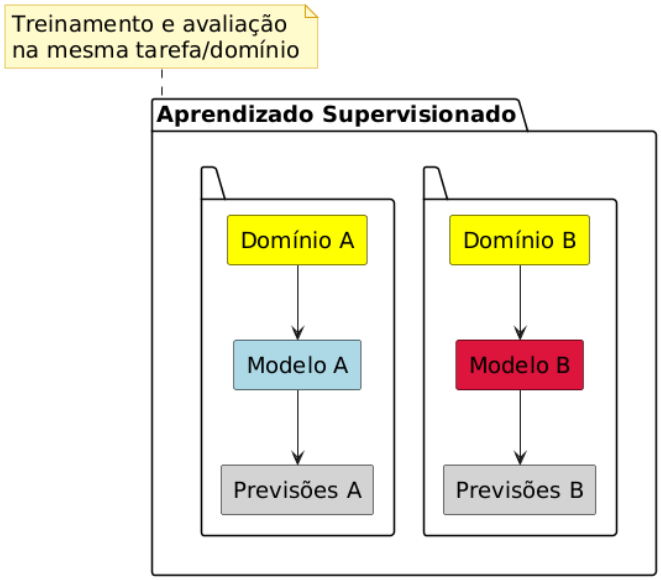
Fonte: Autoria própria (2025).

2.3 Avanços recentes em PLN

Como foi apresentado anteriormente, Aggarwal (2017) explica que historicamente as técnicas de detecção de *outliers* evoluíram em quatro etapas principais. A primeira inclui métodos estatísticos clássicos, baseados em média, desvio-padrão e limiares. Na segunda etapa surgem os métodos *instance-based*, que usam distâncias ou densidade local. A terceira etapa introduz os modelos de generalização explícita, como *One-Class SVM* e *Isolation Forest*. Por fim, a quarta etapa marca a chegada dos ***Transformers*** e dos **LLMs** (Modelos de Linguagem de Larga Escala), que incorporam contexto semântico de maneira muito mais eficiente.

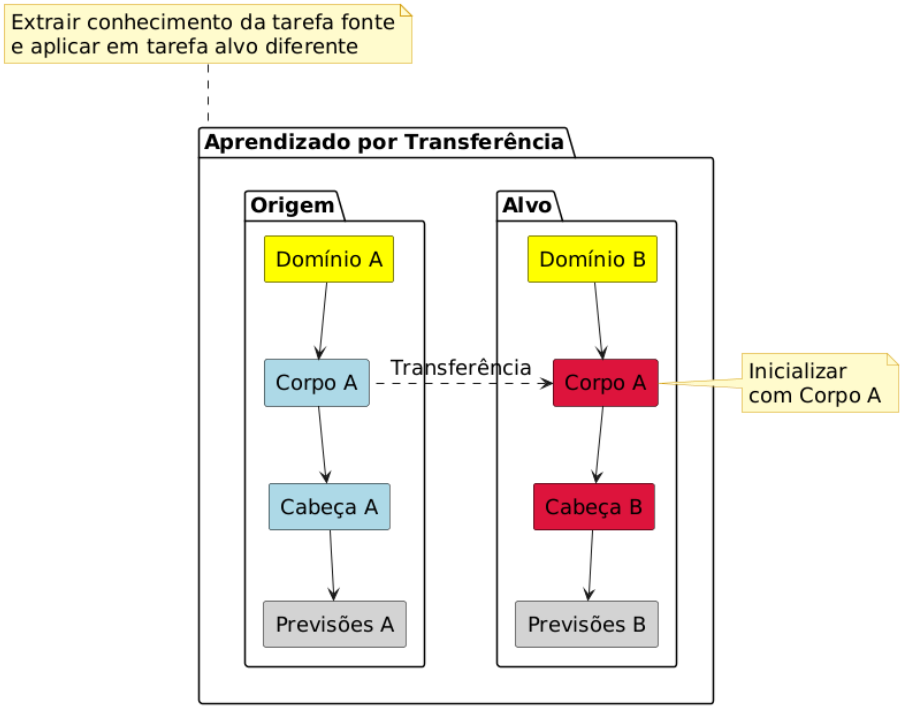
O conceito de *transfer learning* é uma base fundamental da evolução recente no PLN. Essa técnica, originalmente difundida na área de visão computacional, passou a transformar o PLN a partir de 2017, com o surgimento de modelos como o BERT, baseados na arquitetura *Transformer*. Segundo Tunstall, Werra e Wolf (2022), a proposta dos autores consiste em treinar um modelo em uma tarefa ampla (pré-treinamento) e reutilizar esse conhecimento em tarefas específicas, por meio de uma adaptação leve. As Figuras 6 e 7, comparam o fluxo tradicional de aprendizado supervisionado com o fluxo de *transfer learning*. No primeiro, cada tarefa exige o treinamento completo de um modelo. Já no segundo, o corpo do modelo é mantido e apenas a parte final (*head*) é ajustada para a nova tarefa. Esse processo permite obter resultados mais robustos mesmo com quantidades limitadas de dados rotulados.

Figura 6 – Arquitetura de Aprendizado supervisionado



Fonte: Autoria própria (2025).

Figura 7 – Arquitetura Aprendizado por transferência.



Fonte: Autoria própria (2025).

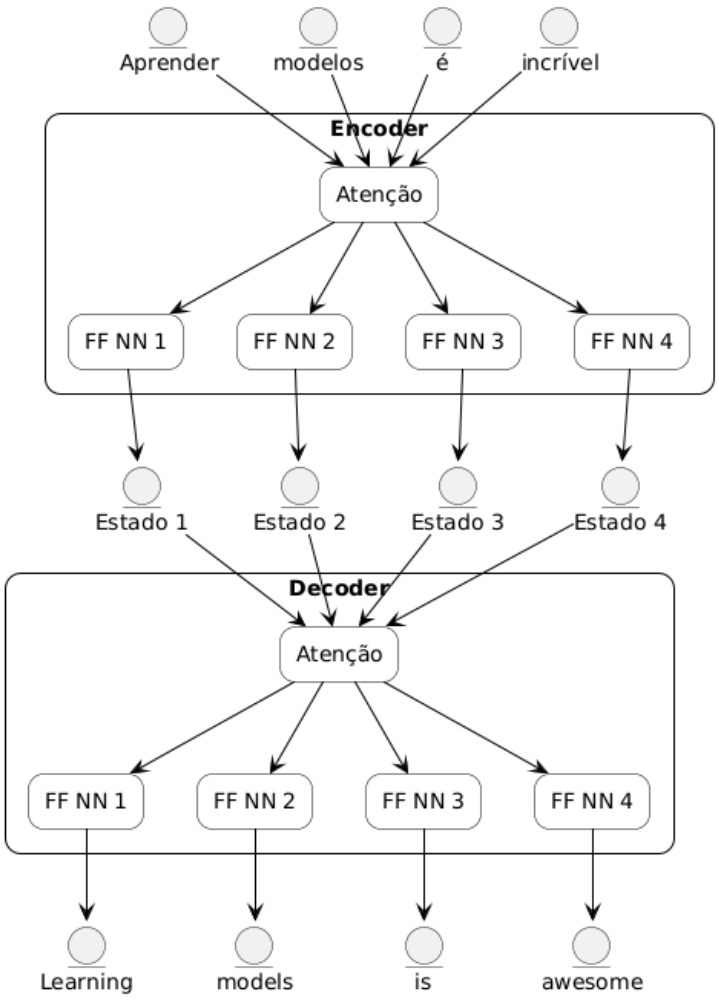
O *Transformer* é a arquitetura que viabilizou esse avanço. Seu funcionamento baseia-se em representações vetoriais (*embeddings*), explicadas na seção anterior, que servem como entrada para o modelo. O *Transformer* adota uma estrutura do tipo *encoder-decoder*, cujo principal mecanismo é o ***self-attention***. Esse mecanismo permite que cada palavra da sequência interaja diretamente com todas as outras, independentemente da posição em que aparecem no texto.

Com isso, o modelo é capaz de gerar representações altamente contextualizadas, superando as limitações dos modelos sequenciais anteriores. Segundo Tunstall, Werra e Wolf (2022):

> "A função do encoder é transformar a sequência de entrada em uma representação numérica, conhecida como último estado oculto. Este estado é então passado para o decoder, que gera a sequência de saída." (TUNSTALL; WERRA; WOLF, 2022)

A Figura 8 apresenta a arquitetura geral de um *Transformer*. Ela possui dois blocos principais: o *encoder* e o *decoder*. Ambos utilizam *self-attention*. O *encoder* processa toda a entrada e gera vetores que capturam relações entre as palavras. O *decoder*, além do *self-attention*, possui um módulo de *cross-attention*, que conecta as representações do *encoder* com o processo de geração da saída. Após o bloco de atenção, cada *token* passa por uma camada de **Feed-Forward Neural Network (FFNN)**. A FFNN é aplicada de forma independente a cada posição da sequência. Ela tem a função de transformar os vetores de atenção em representações mais abstratas, aumentando a capacidade do modelo de capturar padrões não lineares.

Figura 8 – Arquitetura *Encoder-Decoder* (*Transformer*).



Fonte: Autoria própria (2025).

2.3.1 Modelos pré-treinados

A partir de 2018, surgiram modelos fundamentais para a evolução dos *Transformers* aplicados ao Processamento de Linguagem Natural (PLN), como o *Generative Pre-trained Transformer* (GPT) (OPENAI, 2023), o modelo DeepSeek (AI, 2024) e o *Bidirectional Encoder Representations from Transformers* (BERT) (DEVLIN *et al.*, 2018). Esses modelos pertecem ao campo da Inteligência Artificial Generativa, caracterizada pela capacidade de produzir novos dados com base nos padrões aprendidos durante o treinamento, incluindo texto e conteúdo multimídia (TUNSTALL; WERRA; WOLF, 2022). Também conhecidos como *Large Language Models* (LLMs), esses sistemas estatísticos, de acordo com a própria documentação da Google (2025) utilizam o mecanismo de *self-attention* junto ao *transfer learning*. Em consequência, permite pré-treinar modelos em grandes volumes de dados e posteriormente adaptá-los para tarefas específicas.

Particularmente, os modelos generativos como o **DeepSeek** utilizam exclusivamente a arquitetura baseada em *decoder* dos *Transformers*. De forma resumida, seu funcionamento envolve três etapas principais (TUNSTALL; WERRA; WOLF, 2022):

- **Pré-treinamento (*Pre-training*):** onde o modelo é treinado para prever a próxima palavra em uma sequência, com base nas palavras anteriores;
- **Adaptação de domínio (*Domain Adaptation*):** que consiste na adaptação do modelo a um *corpus* específico do domínio desejado;
- **Ajuste fino (*Fine-tuning*):** onde o modelo é ajustado para uma tarefa específica, geralmente adicionando uma camada de classificação na saída.

Por outro lado, o modelo **BERT** utiliza exclusivamente a arquitetura de *encoder* dos Transformers e introduz um método de pré-treinamento inovador denominado *Masked Language Modeling* (MLM). Nesse método, algumas palavras da entrada são ocultadas aleatoriamente, e o modelo deve prevê-las com base no contexto fornecido pelas demais palavras. Essa estratégia permite ao BERT aprender representações bidirecionais mais ricas e sensíveis ao contexto (TUNSTALL; WERRA; WOLF, 2022).

Adicionalmente, em abordagens mais atuais de acordo com Bommasani *et al.* (2022), modelos como BERT, RoBERTa, GPT, T5, PaLM e GPT-4 são denominados modelos fundacionais porque são treinados em uma escala massiva, utilizando dados híbridos da *web*, livros, artigos e outros textos, sem a necessidade de rótulos manuais. Eles são construídos sobre o princípio de aprendizado auto-supervisionado, onde o próprio texto fornece a supervisão para o treinamento. O termo *fundacional* reflete o fato de que esses modelos são usados para diversas tarefas, incluindo classificação de texto, tradução, sumarização, detecção de anomalias, geração de texto e muito mais (BOMMASANI *et al.*, 2022).

2.3.2 Geração Aumentada por Recuperação (RAG)

Com a popularização dos modelos LLM, surge uma alternativa promissora para gerar respostas a partir de diferentes fontes de informação. Tunstall, Werra e Wolf (2022) explicam a abordagem conhecida como *Retrieval-Augmented Generation* (RAG), cuja ideia central consiste em recuperar trechos relevantes de documentos externos e utilizá-los como contexto adicional para a geração textual realizada por um modelo de linguagem pré-treinado. Trata-se de uma estratégia que combina recuperação de informação com geração de linguagem natural, permitindo respostas mais completas e fundamentadas.

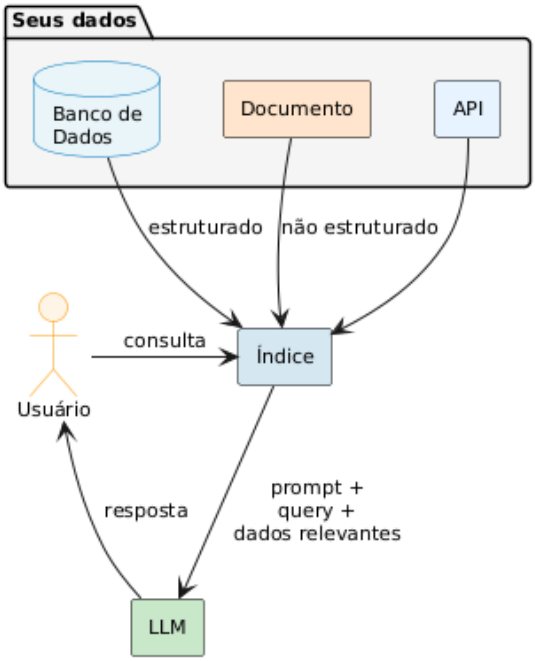
A Figura 9 ilustra o funcionamento geral de um sistema RAG. Inicialmente, são utilizados dados estruturados, não estruturados (documentos) e informações obtidas por meio de APIs. Esses dados são processados e convertidos em representações vetoriais (*embeddings*), sendo posteriormente indexados e organizados em um repositório vetorial.

O processo de classificação inicia quando o usuário envia uma consulta denominada *prompt*. Segundo Tunstall, Werra e Wolf (2022), o *prompt* corresponde ao texto de entrada utilizado para orientar o comportamento do modelo. Na etapa de *Retrieval*, o documento *target* a ser classificado, também é convertido em vetor (*embedding*) e comparado com os documentos *source* indexados através de busca por similaridade (produto interno). O documento *source* mais similar é então recuperado em formato textual.

Na etapa de *Augmentation*, o documento *source* recuperado é combinado com o documento *target* e uma lista de pontuação para formar um contexto para o modelo. Como explicado por Liu *et al.* (2025), no contexto de detecção de novidades, a estrutura de um *prompt* seria: (i) instruções sobre a tarefa do modelo, (ii) critérios de classificação (escala de 0 a 1), e (iii) os textos completos dos documentos a serem comparados. Finalmente, na etapa de *Generation*, o *prompt* estruturado é enviado ao modelo de linguagem de grande escala, que processa o contexto textual e gera uma classificação binária (novidade ou não novidade).

É importante destacar que o LLM recebe e processa apenas informações textuais, não vetores. Os *embeddings* são utilizados exclusivamente na etapa de recuperação por similaridade. Por conseguinte, a decisão final de classificação é realizada pelo modelo de linguagem sobre o conteúdo textual dos documentos. Nesta abordagem baseada em *prompts*, o usuário mantém controle direto sobre o tipo de consulta realizada, os dados utilizados e o formato da resposta esperada.

Figura 9 – Fluxo básico de um sistema RAG.



Fonte: Autoria própria (2025).

É essencial diferenciar o processo de RAG do processo de *fine-tuning*. Conforme destacado por Belcic e Stryker (2024), existe uma diferença fundamental entre ambas as abordagens. O RAG amplia as capacidades de um modelo de linguagem ao conectá-lo a fontes externas de dados, sem modificar seus parâmetros internos. Já o *fine-tuning* retreina o modelo pré-treinado usando um conjunto de dados específico, incorporando conhecimento especializado. Assim, enquanto o RAG fornece atualizações dinâmicas baseadas no conteúdo recuperado, o *fine-tuning* adapta o modelo para um domínio específico.

2.4 Métricas de Avaliação

Para validar tarefas de classificação binária, como a detecção de novidade, é comum utilizar métricas como precisão, revocação (*recall*), F1-*score* e acurácia. Antes do cálculo dessas métricas, é importante compreender a terminologia utilizada. As principais variáveis usadas, segundo a documentação de Scikit-Learn Developers (2024), são:

- **Verdadeiro Positivo (TP):** instâncias positivas corretamente previstas como positivas.
- **Falso Positivo (FP):** instâncias negativas incorretamente previstas como positivas.
- **Falso Negativo (FN):** instâncias positivas incorretamente previstas como negativas.
- **Verdadeiro Negativo (TN):** instâncias negativas corretamente previstas como negativas.

Consequentemente, na Tabela 6, são calculadas outras relações para medir o desempenho das clasificações.

Tabela 6 – Relação entre métricas básicas e suas descrições

Equação	Descrição
$PP = TP + FP$	**Positivos previstos**: total de instâncias que o modelo classificou como positivas, incluindo acertos (TP) e erros (FP).
$PN = TN + FN$	**Negativos previstos**: total de instâncias classificadas como negativas pelo modelo, incluindo acertos (TN) e erros (FN).
$RP = TP + FN$	**Positivos reais**: total de instâncias que realmente pertencem à classe positiva, independentemente da previsão do modelo.
$RN = TN + FP$	**Negativos reais**: total de instâncias pertencentes à classe negativa no conjunto de dados.
$N = TP + TN + FP + FN$	Número total de instâncias avaliadas pelo modelo.

Fonte: Autoria própria (2025).

A **precisão** mede a proporção de amostras realmente positivas entre todas aquelas classificadas como positivas. Segundo a definição da biblioteca *Scikit-learn*, essa métrica expressa a capacidade do modelo de evitar classificações positivas incorretas (Scikit-Learn Developers, 2024):

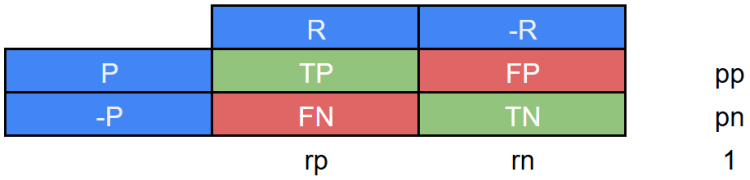
$$\text{Precisão} = \frac{TP}{TP + FP} \tag{1}$$

O *recall* representa o quanto o modelo é capaz de identificar corretamente os casos positivos. Segundo Powers (POWERS, 2011), essa medida avalia a cobertura das amostras relevantes capturadas pela regra de previsão positiva:

$$\text{recall} = \frac{TP}{TP + FN} \tag{2}$$

A Figura 10 mostra uma tabela de contingência binária. O quadrante A (verde - acerto) representa os verdadeiros positivos, ou seja, casos positivos corretamente identificados. O quadrante B (vermelho - erro) representa os falsos positivos, que ocorrem quando o modelo classifica como positivo algo que na verdade é negativo. O quadrante C (vermelho - erro) mostra os falsos negativos, que são casos positivos que foram incorretamente classificados como negativos. Por fim, o quadrante D (verde - acerto) representa os verdadeiros negativos, ou seja, as instâncias negativas corretamente classificadas como tal.

Figura 10 – Exemplo matriz de confusão binária.



Fonte: Autoria própria (2025).

Para combinar precisão e *recall* em uma única métrica, utiliza-se o **F1-score**, que é a média harmônica entre elas. Essa métrica é útil em cenários com classes desbalanceadas. A fórmula pode ser representada conforme descrito por Taha e Hanbury (2015):

$$\text{F1-score} = \frac{2TP}{2TP + FP + FN} \tag{3}$$

A **acurácia** mede a proporção de previsões corretas em relação ao total de amostras. Essa métrica é apropriada quando as classes estão balanceadas (Scikit-Learn Developers, 2024):

$$\text{Acurácia} = \frac{TP + TN}{TP + FP + TN + FN} \tag{4}$$

3 TRABALHOS RELACIONADOS

Particularmente, os avanços nos modelos fundacionais são relevantes para tarefas de detecção de novidade, pois permitem gerar representações semânticas contextuais. Embora esses grandes modelos dominem aplicações como tradução, sumarização e resposta a perguntas, estudos como Ghosal *et al.* (2018) e Nair (2023) demonstram que, na detecção de novidade documental, abordagens tradicionais supervisionadas e não supervisionadas continuam competitivas, especialmente em cenários com escassez de dados rotulados.

A principal contribuição de Ghosal *et al.* (2018) foi a criação do conjunto de dados ***TAP-DLND 1.0***, amplamente utilizado como *benchmark* na área. O conjunto de dados é composto por 6.109 notícias distribuídas em 10 temas (como política, esportes e negócios), contendo aproximadamente **50% de documentos rotulados como *novel*** e **50% como *non-novel***. Cada evento é estruturado em dois subconjuntos, explicados na Tabela 7.

Tabela 7 – Estrutura do conjunto de dados *TAP-DLND 1.0*

Subconjunto	Descrição
Source	Contém três notícias de referência que descrevem o evento base. Essas notícias formam o conhecimento inicial a partir do qual a novidade é avaliada.
Target	Inclui dezenas de notícias adicionais relacionadas ao mesmo evento. Cada uma deve ser classificada como Novidade ou Não novidade.

Fonte: Adaptado de Ghosal *et al.* (2018).

Além da construção do *corpus*, os autores propuseram uma abordagem **supervisionada** dividida em duas etapas principais. Na primeira, foi criado um novo conjunto de dados derivado do *TAP-DLND 1.0*, construído a partir das diferenças entre documentos *source* e *target*. Para cada par de documentos, foram extraídas diversas características que buscavam quantificar o grau de novidade lexical e semântica. As principais características utilizadas incluem:

- **Paragraph Vector**: diferença entre vetores calculada pelo cosseno;
- **Concept Centrality**: diferença de centralidade semântica entre documentos;
- **N-grams**: conjuntos de 2, 3 e 8 termos consecutivos;
- **Named Entities e Keywords Match**: correspondência entre entidades nomeadas e palavras-chave;
- **New Word Count**: contagem de palavras novas;
- **Divergence**: medida de divergência entre distribuições de termos.

Trabalhar com um novo conjunto de dados permitiu abordar o problema como uma **tarefa supervisionada**. Já na segunda etapa, foi utilizado um classificador *Random Forest*, treinado com validação cruzada (*cross-validation*) de 10 *folds*, utilizando as características extraídas anteriormente. Os dados possuíam rótulos binários, e a tarefa consistia em distinguir entre documentos de novidade e de não novidade. Essa abordagem, baseada no cálculo de similaridade por cosseno, alcançou acurácia de **79,2%**, evidenciando que, mesmo com um classificador supervisionado robusto, a tarefa de identificar novidade documental é complexa. Em muitos casos, características puramente lexicais, como frequência de palavras ou *n-grams*, não são suficientemente expressivas para capturar diferenças semânticas relevantes.

Após a publicação do artigo do TAP-DLND 1.0 em 2018, que lançou a primeira versão do *dataset* de *benchmark*, Ghosal *et al.* (2018) deram continuidade ao avanço metodológico com duas contribuições importantes. A primeira, também de 2018, foi o trabalho *"Novelty Goes Deep: A Deep Neural Solution to Document-Level Novelty Detection"* (GHOSAL *et al*., 2018), no qual os autores propõem a arquitetura RDV-CNN, um modelo profundo baseado na construção de vetores diferenciais entre documentos (*Representation Difference Vectors*) combinados com uma rede convolucional hierárquica. Essa abordagem superou resultados anteriores, alcançando *F1-scores* superiores a **80,0%** no *corpus* TAP-DLND 1.0. Posteriormente, em 2021, foi publicado o trabalho *"Is Your Document Novel? Let Attention Guide You"* (GHOSAL *et al*., 2020), no qual os autores introduzem um modelo que utiliza um mecanismo de atenção para identificar partes do texto com maior importância, redundância ou contradição informacional. Essa versão apresentou desempenho superior, com F1-scores superior a **83,0%**, consolidando o uso de atenção e inferência semântica como estratégias eficazes para a tarefa.

Mais recentemente, Nair (2023) propôs um método completamente não supervisionado baseado em coocorrência e associação de palavras, sem o uso de *embeddings* nem de aprendizado profundo. No *benchmark TAP-DLND 1.0*, o autor aplicou um pré-processamento tradicional (remoção de *stopwords*, tokenização e lematização) e criou uma *Term-Document Matrix* (TDM) com esparcidade reduzida (limite 0.99). A partir disso, selecionou os seis termos de maior frequência, considerados palavras chave, e comparou-os com os termos do conjunto *source*. Dependendo da correlação entre esses termos, cada documento foi classificado como novidade ou não novidade. O sistema foi avaliado, obtendo um F1-score de **73,8%** utilizando apenas três documentos de referência por evento. Esse resultado demonstra que métodos não supervisionados podem alcançar desempenho comparável a modelos supervisionados, com custo computacional significativamente menor.

Além disso, existem pesquisas que exploram o uso de grandes modelos fundacionais, como GPT-3.5, GPT-4 e LLaMA-2, para detectar novidades em textos científicos, com destaque para o uso de *embeddings* sem ajuste fino. Liu *et al.* (2025) realizaram experimentos demonstrando que o GPT-4, mesmo sem treinamento adicional, superou modelos tradicionais e supervisionados, atingindo precisão em torno de a 80% em benchmarks de identificação de ideias emergentes. No mesmo estudo, Liu *et al.* (2025) apresentam um sistema baseado em RAG para

detecção de novidades e descrevem como configurá-lo por meio de *prompting* especializado. Eles também propõem uma escala de pontuação [0, 0.1, 0.3, 0.5, 0.7 e 1] para classificar documentos conforme o grau de novidade interpretado pelo modelo fundacional. Adicionalmente, os autores sugerem testar modelos mais recentes, como o da DeepSeek, devido ao bom desempenho, maior capacidade interpretativa e custos reduzidos da API. Em conjunto, esses estudos evidenciam a complementaridade entre abordagens léxicas não supervisionadas e arquiteturas modernas baseadas em representações semânticas contextualizadas.

Em síntese, a literatura sobre detecção de novidade documental apresenta uma evolução metodológica significativa. Inicialmente, Ghosal *et al.* (2018) propuseram abordagens supervisionadas clássicas baseadas em características lexicais e semânticas. Posteriormente, surgiram arquiteturas de aprendizado profundo com mecanismos de atenção, descritas por Ghosal *et al.* (2018) e Ghosal *et al.* (2020). Mais recentemente, métodos não supervisionados fundamentados em coocorrência de termos foram apresentados por Nair (2023). Por fim, Liu *et al.* (2025) exploraram o uso de modelos fundacionais com arquiteturas RAG para a tarefa. Os resultados demonstram que abordagens não supervisionadas mantêm-se competitivas, especialmente em cenários com escassez de dados rotulados. No entanto, permanecem lacunas na integração de métodos clássicos de detecção de outliers com técnicas modernas de recuperação aumentada. Por esse motivo, este trabalho visa explorar abordagens completamente não supervisionadas, combinando detectores clássicos e arquiteturas RAG para identificação de novidades documentais.

4 MATERIAIS E MÉTODOS

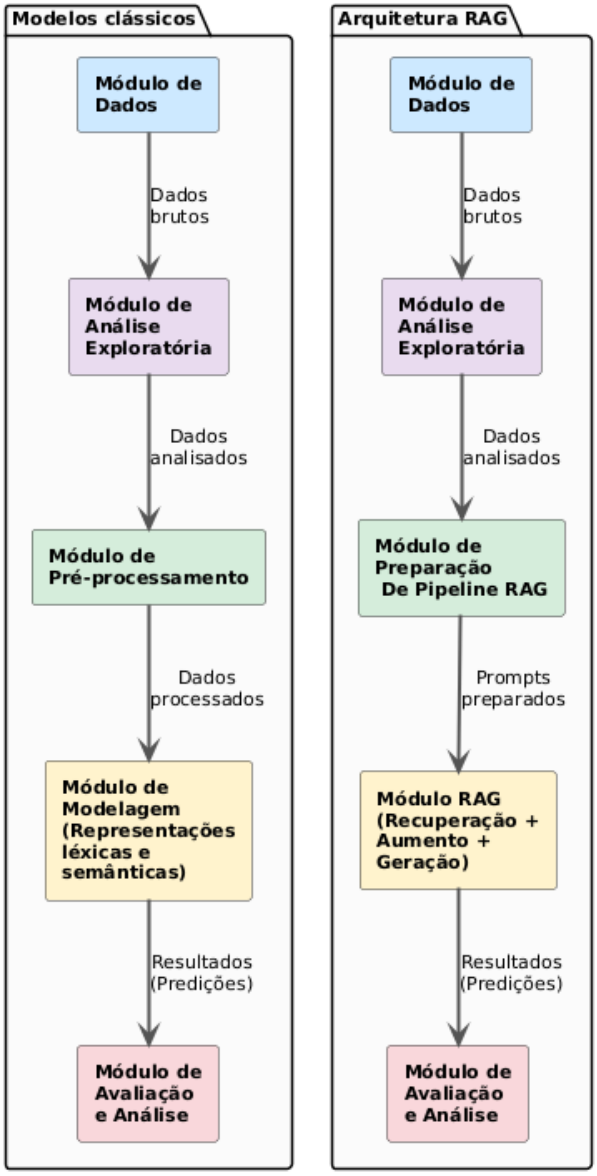
A ênfase deste capítulo está em apresentar a estrutura do trabalho desenvolvido sob métodos de aprendizado não supervisionado, organizando cada etapa de forma clara e detalhada. O trabalho foi dividido em duas abordagens, ambas avaliadas sobre o mesmo conjunto de dados. Vale ressaltar que as *features* do conjunto de dados ***source*** (três documentos por evento) foram utilizadas exclusivamente para treinamento, mantendo a abordagem não supervisionada, ou seja, sem acesso às etiquetas durante o processo de modelagem. Já as etiquetas da *feature* DLA do conjunto de dados ***target*** foram empregadas unicamente na etapa de análise dos resultados, para fins de avaliação do desempenho dos modelos. A primeira abordagem consiste em métodos de classificação clássicos como LOF, *Elliptical Envelope* e *Isolation Forest*. A segunda abordagem emprega uma técnica amplamente utilizada na atualidade, a arquitetura RAG. Nestes métodos de classificação não supervisionada, o problema de detecção de novidade é tratado como um caso de detecção de *outliers*, conforme descrito por Aggarwal (2017).

4.1 Disposição do projeto

A primeira abordagem da Figura 11 representa, de forma modular, a abordagem não supervisionada dividida em cinco etapas principais. O primeiro módulo corresponde ao **conjunto de dados**, responsável por armazenar e organizar as informações que serão utilizadas durante o desenvolvimento. Em seguida, o segundo módulo contempla a **análise exploratória**, utilizada para compreender o comportamento dos dados. A partir desse conjunto, tem-se o terceiro módulo, dedicado ao **pré-processamento dos dados**, no qual são aplicadas as técnicas descritas no capítulo anterior, como limpeza, tokenização, transformação textual e vetorização. O quarto módulo refere-se à **aplicação dos algoritmos de detecção de novidade**, etapa na qual os modelos são empregados para identificar padrões ou informações que representem novidade. Por fim, o quinto módulo corresponde à etapa de **análise dos resultados**, que visa apresentar as saídas geradas, bem como fornecer interpretações e avaliações dos resultados obtidos.

A segunda abordagem da Figura 11 é baseada em RAG, seguindo uma estrutura modular semelhante à anterior. A primeira etapa corresponde à **obtenção do conjunto de dados**. A segunda etapa contempla a **análise exploratória**, utilizada para compreender o comportamento e as características dos dados que serão necessárias para construir os *prompts* corretos. A terceira etapa refere-se à **Preparação do *Pipeline RAG***, na qual são definidas as APIs, versões do modelo de linguagem e especificação dos *prompts* de entradas e saídas desejadas para o LLM. Em seguida, o **módulo RAG** integra os dados previamente analisados ao modelo de linguagem, realizando três operações principais: recuperação de documentos relevantes, aumento do contexto através de engenharia de *prompts* e geração de respostas. Por fim, na quinta etapa ocorre a **avaliação e análise dos resultados do RAG**, onde são calculadas as métricas de desempenho e verificada a qualidade das classificações produzidas pelo modelo.

Figura 11 – Sequência operacional das abordagens não supervisionadas.



Fonte: Autoria própria (2025).

4.2 Detalhamento do banco de dados utilizado

Para aplicar as abordagens, foi selecionado como base o conjunto **TAP-DLND 1.0** (GHOSAL *et al.*, 2018), considerado um dos principais *benchmarks* para tarefas de detecção de novidade documental. Este foi construído a partir de notícias jornalísticas em *inglês*, organizadas em eventos temáticos. A estrutura do conjunto é dividida em duas pastas principais: *source*, que reúne documentos de referência, e *target*, composta por documentos que devem ser avaliados quanto ao seu grau de novidade em relação aos textos base. A Tabela 8 mostra de forma simplificada a disposição do *Dataframe*. A principal diferença entre os dois conjuntos de dados reside nas etiquetas. O conjunto *source* **não contém dados rotulados**, contém apenas **três** documentos para cada evento e serve como conjunto base para comparação. Por outro lado, o

conjunto de dados *target* possui a coluna de dados rotulados do tipo *novel* ou *non-novel*. Um exemplo de instância do *corpus* pode ser visualizado no **Anexo A**.

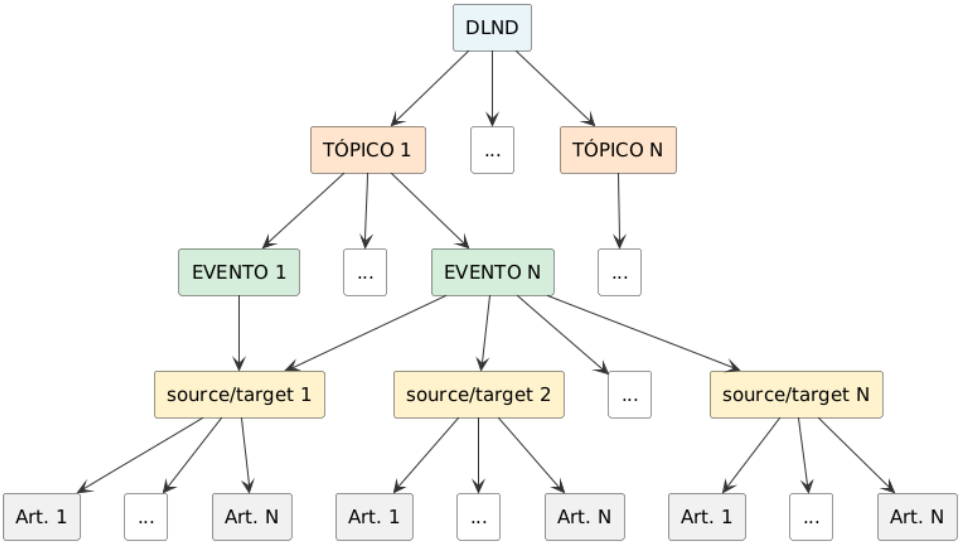
Tabela 8 – Composição do *Dataframe*

Característica	Tipo
`category`	Categórica (texto)
`event_id`	Numérica (inteira)
`news_id`	Identificador (texto)
`content`	Texto (livre)
`is_source`	Binária (0/1)
`DOP`	Temporal (data/ano)
`publisher`	Categórica (texto)
`title`	Texto (curto)
`eventid`	Identificador (texto)
`eventname`	Categórica (texto)
`topic`	Categórica (texto)
`sentence`	Numérica (inteira)
`words`	Numérica (inteira)
`sourceid`	Identificador (texto)
`DLA`	Binária (0/1)
`SLNS`	Numérica (contínua)

Fonte: Autoria própria (2025).

A Figura 12 exemplifica de melhor forma a estrutura do conjunto de dados. Desta forma, está dividido em 3 níveis de pastas, sendo o primeiro nível para categorias. O **TAP-DLND 1.0** está distribuído em 10 categorias temáticas. No segundo nível, para cada categoria existem diferentes eventos, totalizando 223 eventos. Finalmente, no terceiro nível, para cada evento o *dataset* apresenta a divisão de *source* e *target*, totalizando 6.109 documentos anotados. A Tabela 9 mostra as 10 categorias e as respectivas divisões: Novidade (N) com um total de 2.804 registros e Não Novidade (NN) com um total de 2.631 registros.

Figura 12 – Estrutura Hierárquica do *corpus* DLND.



Fonte: Autoria própria (2025). Adaptado de Ghosal *et al.* (2018).

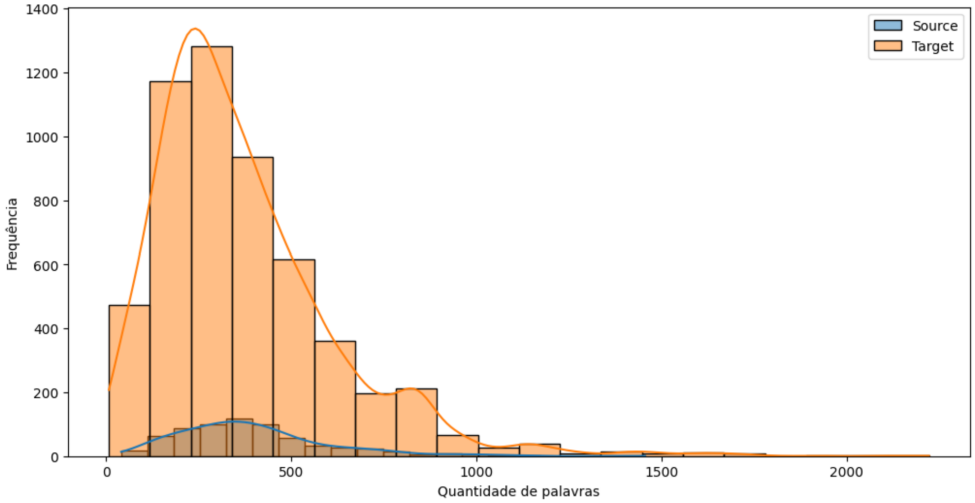
Tabela 9 – Distribuição de eventos e documentos por categoria

Categoria	# Eventos	# N	# NN
ACC (Acidentes)	10	231	272
PLT (Política)	97	669	685
BUS (Negócios)	35	202	264
ART (Arte)	21	397	258
CRM (Crime)	10	237	174
NAT (Natureza)	10	87	250
TER (Terrorismo)	18	255	468
GOV (Governo)	15	405	219
SPT (Esportes)	2	39	51
SOC (Social)	5	214	63

Fonte: Autoria própria (2025). Adaptado de Ghosal *et al.* (2018).

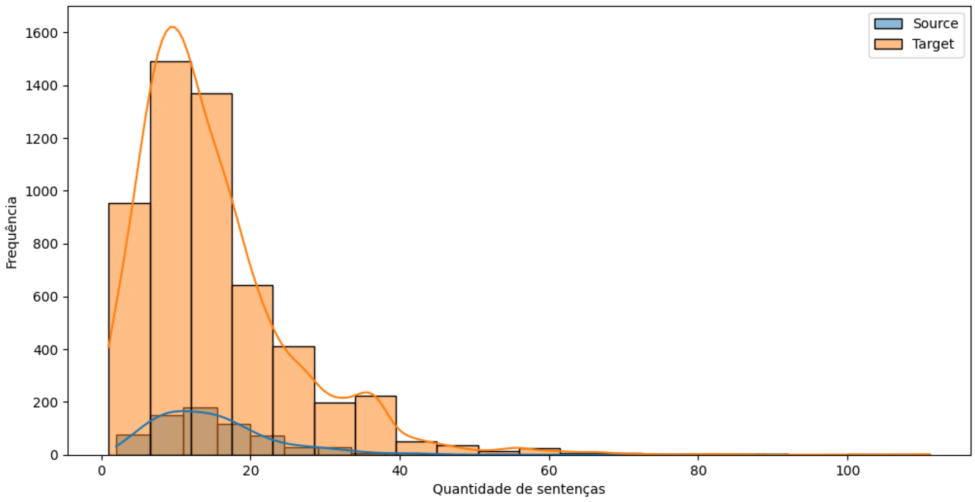
As Figuras 13 e 14 apresentam a distribuição de palavras e sentenças, respectivamente, nos conjuntos *source* e *target*. Observa-se que ambos os conjuntos possuem distribuições assimétricas à esquerda, com a maioria dos documentos concentrada entre 200 e 400 palavras (Figura 13) e entre 10 e 20 sentenças (Figura 14). Nota-se ainda a presença de *outliers* com mais de 1000 palavras ou 40 sentenças, embora representem uma pequena fração do *corpus*.

Figura 13 – Distribuição de palavras.



Fonte: Autoria própria (2025).

Figura 14 – Distribuição de sentenças.



Fonte: Autoria própria (2025).

Segundo Ghosal *et al.* (2018), a contagem de palavras novas NWC (*New Word Count*) é o atributo de maior importância neste conjunto de dados. Assim, a similaridade observada entre as distribuições de *source* e *target* é fundamental para essa análise. As métricas baseadas em sobreposição, como *n-grams* e NWC, são sensíveis ao tamanho do texto (GHOSAL *et al.*, 2018). Se as distribuições fossem discrepantes, tais métricas seriam desiguais. Portanto, o equilíbrio observado sugere que se realize uma interpretação focada nas diferenças lexicais e semânticas.

Para compreender em profundidade o conteúdo dos documentos em cada categoria, foram gerados gráficos de frequência de palavras-chave. O **Apêndice A** detalha de forma gráfica as palavras predominantes por categoria, considerando todos os eventos. A análise dessas visualizações demonstra uma forte correlação temática entre os conjuntos *source* e *target*. As mesmas palavras-chave predominam em ambos os conjuntos, o que indica que os documentos estão relacionados à mesma temática. Contudo, essa sobreposição lexical revela o principal

desafio, a tarefa não é apenas identificar a temática, mas sim distinguir a semântica contextual da novidade.

4.3 Pré-processamento do conjunto de dados

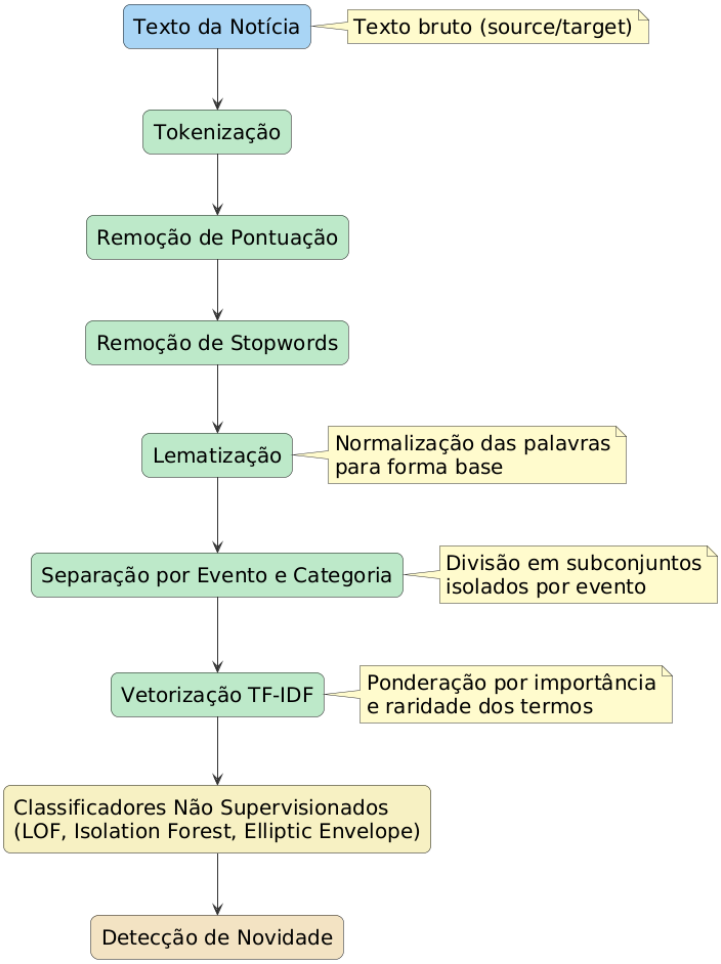
O desenvolvimento deste trabalho foi realizado utilizando a linguagem de programação **Python 3.10.12** (FOUNDATION, 2024), amplamente adotada na área de ciência de dados e aprendizado de máquina. As principais bibliotecas utilizadas incluem **Scikit-learn 1.3.0** (Scikit-Learn Developers, 2024) para implementação dos algoritmos de detecção, **Pandas 2.0.3** (TEAM, 2024) e **NumPy 1.25.2** (AL., 2020) para manipulação e análise de dados, e **Matplotlib 3.7.1** (HUNTER, 2007) e **Seaborn 0.12.2** (WASKOM, 2021) para a geração de gráficos e visualizações. O ambiente de desenvolvimento utilizado foi o **Jupyter Notebook 6.5.4**, que permite organizar o código em blocos, facilitando a execução modular. A seguir, descreve-se o pré-processamento detalhado realizado para cada abordagem.

4.3.1 Pré-processamento na abordagem não supervisionada

Após a análise exploratória dos dados, é necessário estruturar o processamento na abordagem não supervisionada. Nesta abordagem, o foco está na classificação baseada nas características léxicas e semânticas do texto. Portanto, assim como na abordagem RAG, foram removidas colunas que não contribuem diretamente para a análise textual, tais como `title`, `DOP`, `publisher`, `eventid`, `eventname`, `SLNS` e `sourceid`, mantendo apenas o conteúdo textual da coluna **`content`** para o processamento.

A Figura 15 apresenta o pipeline de processamento adotado, aplicável aos três classificadores utilizados: *Local Outlier Factor* (LOF), *Isolation Forest* e *Elliptic Envelope*. Dessa forma, é possível trabalhar com esses dados de maneira estruturada, considerando que o conjunto *target* será utilizado como base de avaliação, contendo uma coluna que indica se o artigo é considerado **novidade** ou **não novidade**.

Figura 15 – Pipeline de processamento da base de dados.



Fonte: Autoria própria (2025).

O *pipeline* consiste na realização do pré-processamento para cada documento de notícia, que inclui procedimentos como limpeza textual, *tokenização* e vetorização dos dados. Para a primeira etapa, foi *tokenizado* o conteúdo e eliminada a pontuação do texto, para posteriormente remover as *stopwords*. Depois, o texto foi lematizado, deixando todas as palavras no mesmo nível base. Após essas transformações, o texto está preparado para ser convertido em representação numérica (vetores).

A escolha do tipo de vetorização e pré-processamento depende diretamente da abordagem adotada. Para esta abordagem, foi selecionada a vetorização usando TF-IDF, em vez de *Bag of Words* (BoW), pois o TF-IDF pondera a importância das palavras considerando não apenas sua frequência no documento, mas também sua raridade no *corpus* (GRUS, 2019). Essa característica é fundamental para a detecção de novidade, pois termos únicos ou pouco frequentes em um conjunto de documentos podem indicar informações novas, enquanto palavras comuns e repetitivas tendem a representar conteúdo redundante.

Um aspecto fundamental da preparação deste conjunto de dados é garantir que cada categoria e evento sejam tratados como conjuntos isolados durante o treinamento. Misturar documentos de diferentes eventos ou categorias comprometeria a comparação, pois notícias

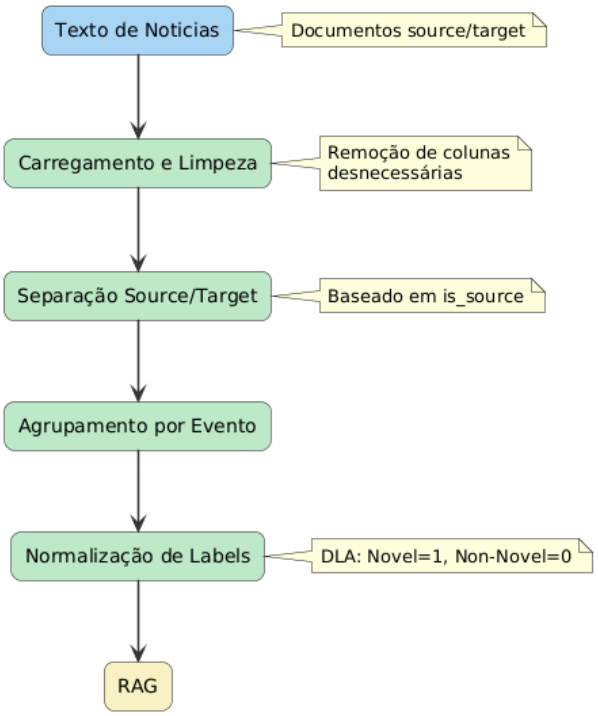
de eventos distintos dentro da mesma categoria (por exemplo, futebol e luta livre na categoria *Sports*) apresentam diferenças léxicas e semânticas significativas que poderiam ser erroneamente identificadas como novidade. Para solucionar esta limitação, as instâncias foram divididas recursivamente em subconjuntos segundo cada evento, garantindo que apenas documentos do mesmo contexto sejam comparados. Após essa divisão, os dados são direcionados para cada classificador, cujos parâmetros e configurações específicas são detalhados nas seções seguintes.

4.3.2 Preparação do *Pipeline* RAG

A abordagem baseada em RAG requer um pré-processamento distinto das técnicas não supervisionadas, pois utiliza modelos de linguagem ou LLMs que processam texto em linguagem natural. Assim, o *pipeline* de preparação foca na organização estrutural dos dados, mantendo o conteúdo textual íntegro para preservar o contexto semântico necessário ao modelo.

A Figura 16 apresenta o fluxo dos dados, partindo do *corpus*, que foi carregado utilizando o `corpusParser`. Em seguida, foram removidas colunas desnecessárias para esta abordagem, tais como `title`, `DOP`, `publisher`, `eventid`, `eventname`, `SLNS` e `sourceid`. Diferentemente da abordagem não supervisionada, o conteúdo textual completo armazenado na coluna `content` foi preservado sem aplicar tokenização, lematização ou remoção de *stopwords*.

Figura 16 – Pipeline de processamento da base de dados (RAG).



Fonte: Autoria própria (2025).

Assim como na abordagem não supervisionada, os documentos foram separados em conjuntos *source* e *target* baseando-se no atributo `is_source`. Posteriormente, esses conjuntos foram agrupados, gerando subconjuntos onde cada elemento corresponde a um evento específico. As etiquetas de novidade na coluna `DLA` foram convertidas para o formato binário, onde documentos classificados como “Novel” receberam o valor 1, e os demais (“Non-Novel”) receberam o valor 0. Dessa forma, os dados ficam preparados para as três etapas principais do RAG: *Retrieval* (recuperação do documento *source* mais similar), *Augmentation* (construção do prompt contextualizado) e *Generation* (classificação pelo LLM), que serão detalhadas nas próximas seções.

4.4 Metodologia para preparação dos classificadores

A preparação dos classificadores é a fase que antecede a geração automática de informações, e é onde o modelo é configurado para realizar predições. Os principais modelos de classificação não supervisionada que serão testados são *Local Outlier Factor* (LOF), *Isolation Forest*, *Elliptic Envelope* e a arquitetura RAG. Esta última abordagem RAG, usará o modelo BERT (DEVLIN *et al.*, 2018) para gerar *embeddings* e o LLM DeepSeek (AI, 2024) para a classificação. Para realizar o processamento conforme descrito anteriormente, os dados usados para a classificação serão unicamente os da coluna `content`, devido à finalidade de comparar os documentos segundo sua composição léxica e semântica.

A implementação dos modelos segue as diretrizes da documentação oficial, onde é necessário compreender e configurar corretamente os parâmetros de cada algoritmo. Os modelos clásicos estão disponíveis na biblioteca **Scikit-learn** (Scikit-Learn Developers, 2024), enquanto o modelo BERT e os recursos para embeddings estão disponíveis no Hugging Face (Hugging Face, 2023), e a API do LLM DeepSeek está documentada em (AI, 2024). Um aspecto fundamental que deve ser realizado na abordagem não supervisionada é ativar corretamente o modo de *novelty detection*, que ajusta o comportamento dos modelos de classificação para avaliar novos dados em relação ao conjunto de referência.

4.4.1 Preparação dos classificadores não supervisionados

Para os três classificadores utilizados (*LOF*, *Isolation Forest* e *Elliptic Envelope*), foi implementada uma função de busca de hiperparâmetros sobre todas as combinações possíveis dos parâmetros. Os valores foram escolhidos a partir da recomendação da documentação no site Scikit-Learn Developers (2024). Esta abordagem sistemática permitiu identificar as configurações ótimas para cada classificador. Além disso, foram adotadas as seguintes práticas comuns:

- Cada classificador foi treinado individualmente para cada evento.

- Os documentos foram transformados em vetores numéricos utilizando TF-IDF com diferentes valores de `ngram_range`.
- Todos os modelos foram configurados com `novelty=True` e com uma contaminação específica `contamination`, que, segundo Scikit-Learn Developers (2024), é a proporção esperada de *outliers* no conjunto de dados.
- Foram calculadas métricas de desempenho (*F1-score*, precisão, *recall*).

4.4.1.1 Local Outlier Factor (LOF)

O LOF detecta *outliers* baseando-se na densidade local dos pontos de dados (BREUNIG *et al.*, 2000) . Para sua configuração, foram explorados os seguintes hiperparâmetros:

- `n_neighbors`: Número de vizinhos considerados para calcular a densidade local. Valores testados: [1, 2, **3**].
- `metric`: Métrica de distância utilizada. Foram avaliadas: cityblock, **cosine**, euclidean e **manhattan**.
- `contamination`: Proporção esperada de anomalias. Valores testados: [0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5].
- `max_features`: Número máximo de características na vetorização TF-IDF. Valores testados: [5, 10, 20, 50, **100**, 1000].
- `ngram_range`: Alcance de n-gramas. Configurações testadas: (1,2) e **(1,3)**.

4.4.1.2 Isolation Forest

O Isolation Forest detecta anomalias isolando observações através de partições aleatórias (LIU; TING; ZHOU, 2008). Os hiperparâmetros explorados foram:

- `n_estimators`: Número de árvores de isolamento. Valores testados: [**50**, 100, 200, 300].
- `max_samples`: Número de amostras para treinar cada árvore. Configurado como 'auto'.
- `contamination`: Valores testados: [**0.05**, 0.1, 0.2, 0.3, 0.4].
- `max_features`: Valores testados: [**100**, 500, 1000, 2000].
- `ngram_range`: Configurações testadas: (1,1), (1,2) e **(1,3)**.

4.4.1.3 Elliptic Envelope

O Elliptic Envelope assume que os dados seguem uma distribuição gaussiana e detecta anomalias que se afastam dessa distribuição (Scikit-Learn Developers, 2024). Os hiperparâmetros configurados foram:

- `contamination`: Valores testados: [**0.05**].
- `max_features`: Valores testados: [**100**, 500].
- `ngram_range`: Configurações testadas: **(1,1)**, (1,2) e (1,3).

4.4.2 Preparação do RAG

Ao contrário das abordagens baseadas em vetorização TF-IDF, este *pipeline* RAG requer uma preparação distinta. O RAG foi implementado seguindo as três etapas descritas na Seção 2.3.2: *Retrieval*, *Augmentation* e *Generation*.

Na etapa de *Retrieval*, foi selecionado o modelo de *embeddings* BAAI/bge-large-en-v1.5 (AI, 2024), que processa o texto bruto e gera representações vetoriais densas de 1024 dimensões, capturando relações semânticas profundas. Com essas representações semânticas, foi realizada uma busca entre o conjunto de dados *source* e *target* para identificar qual notícia possui maior similaridade com cada documento *target*. Os *embeddings* foram armazenados em índices configurados para buscar por produto interno, permitindo recuperar o documento *source* mais relevante. Esta etapa é importante para diminuir o tempo e o custo de processamento nas etapas posteriores.

Na etapa de *Augmentation*, foi utilizado um *prompt* baseado no artigo de Liu *et al.* (2025), onde o autor obteve bons resultados considerando o uso de RAG para detecção de novidade. Com a diferença que, neste trabalho, o próprio LLM realiza a classificação baseado em uma rubrica de pontuação. O *prompt* utilizado encontra-se na Tabela 10. Este está dividido em três componentes: *System*, que indica como o modelo deve se comportar (LIU *et al.*, 2025); *User*, onde são fornecidos os critérios de classificação de forma específica e os documentos a serem comparados; e *Output*, que define o formato que o LLM deve retornar.

Tabela 10 – Estrutura do prompt utilizado para classificação via LLM.

Componente	Conteúdo
System	*You are an [CATEGORY] expert news editor and analyst. Your task is to compare a "new research idea" (the TARGET document) against an "existing research idea" (the SOURCE document) and classify its novelty.*
User	**Clarified Scoring Criteria:** *0.0 – No Novelty (Identical/Reworded): The TARGET is a direct copy or rephrase of the SOURCE.* *0.3 – Low Novelty (Subset/Trivial): The TARGET is a subset of the SOURCE or adds only trivial details.* *0.4 – Moderate-Low Novelty (Same Story): The TARGET shares the exact same central story but adds slightly new context.* *0.6 – Moderate-High Novelty (New Development): The TARGET reports on a new, significant development of the original story.* *1.0 – Very High Novelty (Distinct Story): The TARGET is an entirely distinct and unrelated story.* **Evaluation Instructions:** *If the score is 0.6 or higher, respond "NOVEL". If the score is 0.4 or lower, respond "NO_NOVEL".* *—[INICIO DE SOURCE]— [source_text] —[FIN DE SOURCE]—* *—[INICIO DE TARGET]— [target_text] —[FIN DE TARGET]—*
Output	*NOVEL* ou *NO_NOVEL*

Fonte: Adaptado de Liu *et al.* (2025).

Por último, a etapa de *Generation*, o *prompt* construído foi enviado ao LLM DeepSeek (AI, 2024) usando API, com as configurações de `model="deepseek-chat"`, `temperature=0.0` e `max_tokens=10`. Aqui, o LLM analisa o contexto fornecido e retorna sua classificação no formato especificado Liu *et al.* (2025). A resposta é processada para extrair o resultado (1 para novidade, 0 para não novidade) e armazenado para avaliação posterior. Este processo foi repetido para todos os documentos *target* de cada evento.

4.5 Planejamento Experimental

Nesta seção, descreve-se o planejamento experimental conduzido para avaliar as abordagens de detecção de novidade propostas. São apresentadas as métricas de desempenho desde a perspectiva quantitativa, o protocolo de avaliação baseado no estado da arte e os *baselines* de referência como parâmetro para análise dos resultados deste trabalho.

4.5.1 Avaliação quantitativa

A avaliação quantitativa consiste no cálculo das métricas de classificação para cada modelo em cada categoria. Para avaliar o desempenho dos modelos propostos, foram utilizadas métricas descritas na Seção 2.4: *Precision*, *Recall*, *F1-Score* e Acurácia. Essas métricas são amplamente adotadas na literatura de aprendizado de máquina e permitem uma avaliação

abrangente do desempenho dos classificadores. A escolha desses metodos de avaliação se justifica pela natureza binária do problema de detecção de novidade. Da mesma forma, essas métricas possibilitam a comparação com trabalhos anteriores que utilizaram o mesmo conjunto de dados TAP-DLND 1.0 (GHOSAL *et al.*, 2018).

Os resultados são apresentados em três níveis. Sendo utilizado o *F1-Score* como método principal para seleção de hiperparâmetros e comparação entre modelos, pois equilibra precisão e recall, sendo especialmente relevante em cenários onde as classes podem estar desbalanceadas. Assim, os níveis de análise considerados foram:

- **Por evento**: Análise detalhada do desempenho em eventos específicos, quando relevante.
- **Por categoria**: Desempenho médio de cada modelo em cada uma das dez categorias.
- **Global**: Desempenho total considerando todos os documentos de todas as categorias.

4.5.2 *Baselines* para comparação

Para contextualizar os resultados obtidos, esta pesquisa utiliza como *baselines* os resultados de trabalhos anteriores que também utilizaram o conjunto de dados TAP-DLND-1.0 em seus experimentos:

- No ano de 2018, Ghosal *et al.* (2018) testaram diversos classificadores supervisionados utilizando tanto *features* léxicas (n-gramas, NWC) quanto semânticas (embeddings de palavras). Entre os sistemas avaliados encontram-se:
  - Jaccard + LR (baseline do artigo);
  - Set Difference + LR (ZHANG; CALLAN; MINKA, 2002);
  - Geometric Distance + LR (ZHANG; CALLAN; MINKA, 2002);
  - Language Model (KLD) + LR (ZHANG; CALLAN; MINKA, 2002);
  - Novelty (IDF) + LR (KARKALI *et al.*, 2013);
  - Método proposto por Dasgupta e Dey (2016).

  O melhor resultado reportado por Ghosal *et al.* (2018) foi obtido com a abordagem proposta pelos autores, que utiliza Random Forest de forma supervisionada, construindo iterativamente um novo conjunto de dados e aplicando validação (10-fold). Essa abordagem combinou *features* léxicas e semânticas e alcançou um *F1-Score* médio de aproximadamente 0,80.
- Ainda em 2018, Ghosal *et al.* (2018) apresentaram uma segunda contribuição, introduzindo a arquitetura RDV-CNN, que alcançou *F1-Scores* superiores a 80% no *corpus* TAP-DLND-1.0.

- Em trabalhos mais recentes, Nair (2023) exploraram o mesmo conjunto de dados utilizando técnicas de aprendizado não supervisionado, baseadas em coocorrência e associação de palavras, alcançando *F1-Score* aproximado de 73,8%.

A comparação com esses *baselines* permite avaliar se as abordagens propostas nesta pesquisa, classificadores não supervisionados, representam avanços em relação aos métodos supervisionados tradicionais. Considera-se, de forma particular, as vantagens dos métodos não supervisionados por não requererem dados rotulados para treinamento e a capacidade de raciocínio contextual.

5 ANÁLISE E DISCUSSÃO DOS RESULTADOS

Partindo dos métodos avaliados e de suas respectivas representações vetoriais não supervisionadas, simplificadas na Tabela 11, os resultados da aplicação da metodologia proposta para detecção de novidades são apresentados em detalhes nas seções subsequentes. Esses resultados incluem a descrição do banco de dados, o treinamento dos modelos e a execução dos testes quantitativos.

Tabela 11 – Métodos avaliados e representações vetoriais utilizadas

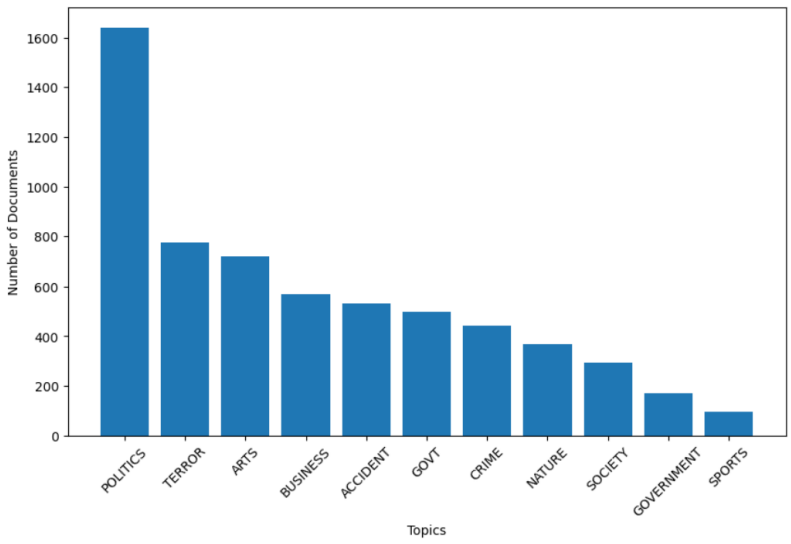
Método	Representação
LOF	TF-IDF
Isolation Forest	TF-IDF
Elliptic Envelope	TF-IDF
RAG	DeepSeek Embeddings + BGE

Fonte: Autoria própria (2025).

5.1 Resultados quantitativos

Considerando as 10 categorias descritas na Figura 17, o conjunto completo totaliza 6.104 documentos, sendo 5.435 documentos *target* e 669 documentos *source* utilizados como base de treinamento. Como descrito na Seção 4.4, foram realizados testes com diferentes métricas de distância, número de vizinhos, níveis de contaminação e variações de n-gramas para ajustar os distintos modelos.

Figura 17 – Número o documents por categoria



Fonte: Autoria própria (2025).

5.1.1 Resultados quantitativos *Elliptic Evelope*

A seguir, foi feita uma análise profunda sobre a resposta do modelo *Elliptic Envelope*, usando como critério comparativo as *baselines* descritas na Subseção 4.5.2. A Tabela 12 mostra que o desempenho da primeira abordagem, particularmente do classificador *Elliptic Envelope*, varia entre as categorias. As métricas com melhor pontuação em F1 médio foram obtidas em *ARTS*, *BUSINESS* e *POLITICS*, onde foram alcançados valores superiores a 75%, indicando que tópicos com maior quantidade de eventos obtiveram resultados superiores. Categorias como *TERROR*, *NATURE* e *ACCIDENT* apresentaram os valores de F1 mais baixos.

Analisando do ponto de vista da quantidade de dados *target*, categorias como *ARTS* com 655 documentos *target* e *TERROR* com 722 documentos obtiveram resultados opostos (F1 de 0,79 e 0,49, respectivamente). Como a quantidade de dados *target* não pode ser considerada isoladamente como parâmetro explicativo, é necessário analisar outros fatores, como a redundância léxica e semântica característica de cada categoria. Essa redundância, particularmente alta em notícias de *TERROR*, pode prejudicar a capacidade do modelo de distinguir novidades reais de reformulações. Para as demais categorias, o modelo apresenta desempenho médio independente da quantidade de eventos ou documentos *target*, com F1 entre 50% e 60%, sugerindo que em *SPORTS*, *SOCIETY*, *GOVT* e *CRIME* o comportamento é próximo ao aleatório, uma vez que as métricas dificilmente superam 70% de F1.

Tabela 12 – Desempenho do *Elliptic Envelope* por categoria.

Categoria	Eventos	F_1 médio (%)	F_1 mediana (%)	Acc. média (%)	Acc. mediana (%)
ACCIDENT	10	58,0	67,0	51,0	53,0
ARTS	21	**79,0**	**80,0**	**73,0**	74,0
BUSINESS	35	**76,0**	67,0	67,0	50,0
CRIME	10	63,0	67,0	56,0	50,0
GOVT	15	69,0	73,0	65,0	62,0
NATURE	10	57,0	66,0	56,0	66,0
POLITICS	96	**75,0**	67,0	69,0	61,0
SOCIETY	5	74,0	**83,0**	67,0	74,0
SPORTS	2	72,0	72,0	63,0	63,0
TERROR	18	49,0	54,0	52,0	50,0

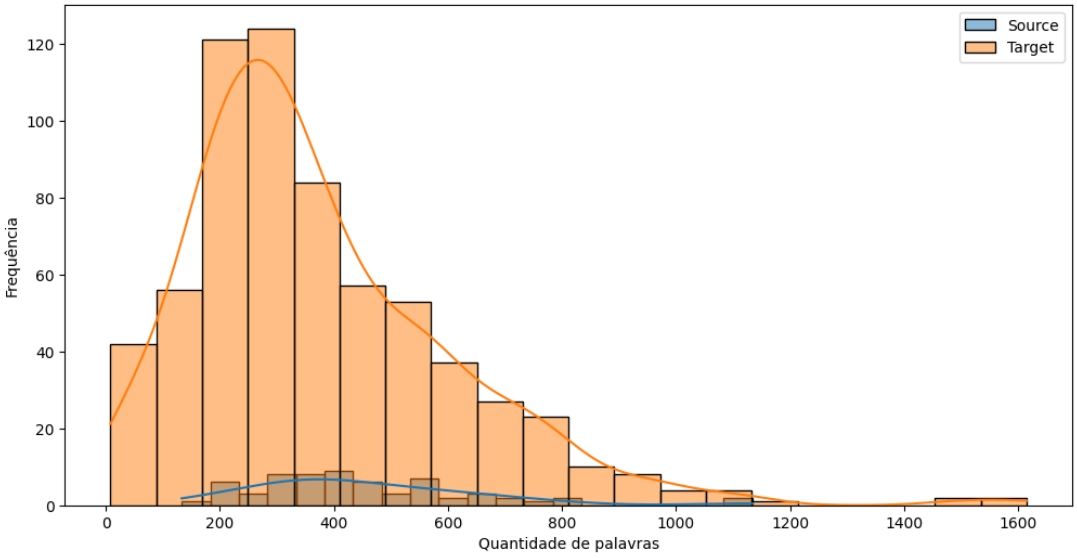
Fonte: Autoria própria (2025).

Com o intuito de analisar a frequência e as características léxicas, as Figuras 18 e 19 mostram a distribuição do comprimento de documentos das duas categorias que apresentaram os resultados mais distantes. De forma geral, os resultados da Figura 18 indicam que a categoria *ARTS* possui uma distribuição aproximadamente gaussiana, com um pico bem definido em torno de 300 palavras. A distribuição de palavras-chave da categoria *ARTS*, em termos de vocabulário, é estável e padronizada, conforme pode ser visualizado no **Anexo A**.

Comparando com a distribuição e o vocabulário da categoria *TERROR* da Figura 19, o classificador *Elliptic Envelope* apresenta maior dificuldade de classificação devido à própria

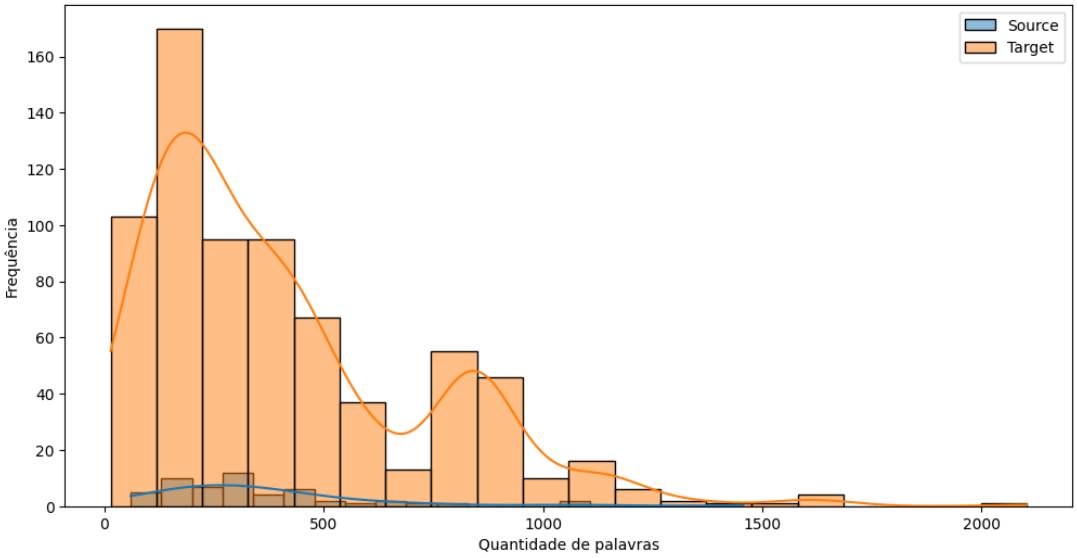
distribuição bimodal, que possui dois picos distintos em aproximadamente 200 e 800 palavras. Isso está alinhado com a respectiva definição do algoritmo, sendo que, segundo Scikit-Learn Developers (2024), o *Elliptic Envelope* é um método que utiliza estimativas de covariância em dados com distribuição gaussiana. Portanto, categorias com distribuições não gaussianas, particularmente *TERROR*, tendem a apresentar desempenho inferior.

Figura 18 – Distribuição do número de palavras na categoria ARTS (*source* vs *target*).



Fonte: Autoria própria (2025).

Figura 19 – Distribuição do número de palavras na categoria TERROR (*source* vs *target*).



Fonte: Autoria própria (2025).

5.1.2 Resultados quantitativos *Isolation Forest*

A Tabela 13 apresenta o desempenho do segundo modelo, o *Isolation Forest*. Os melhores resultados foram obtidos nas categorias *SOCIETY*, *BUSINESS*, *POLITICS* e *ARTS*, onde o F1 médio superou 80%. No entanto, a acurácia não ultrapassou 81% em nenhuma dessas categorias. De forma geral, excluindo esses melhores resultados, o classificador apresentou desempenho consistente nas demais categorias, com valores de F1 entre 64% e 76%.

O algoritmo *Isolation Forest*, como descrito por Liu, Ting e Zhou (2008), baseia-se em partições aleatórias para isolar *outliers*. O princípio fundamental é que *outliers* são mais facilmente isoladas do que observações normais, resultando em caminhos mais curtos nas árvores de decisão. Portanto, quando um conjunto de árvores aleatórias produz caminhos consistentemente mais curtos para determinadas amostras, é altamente provável que essas sejam anomalias. Este mecanismo explica, por exemplo, o desempenho baixo de categorias como *TERROR*. O *Isolation Forest* tem melhor desempenho quando existe uma diferença estrutural clara entre normal e novidade. No entanto, a categoria *TERROR* possui uma distribuição bimodal (conforme Figura 19), e a presença de dois grupos distintos acaba comprometendo a identificação de um padrão.

Tabela 13 – Desempenho do *Isolation Forest* por categoria.

Categoria	Eventos	F_1 médio (%)	F_1 mediana (%)	Acc. média (%)	Acc. mediana (%)
ACCIDENT	10	69,0	76,0	70,0	68,0
ARTS	21	**83,0**	86,0	79,0	81,0
BUSINESS	35	**84,0**	**100,0**	79,0	**100,0**
CRIME	10	72,0	77,0	76,0	81,0
GOVT	15	76,0	76,0	77,0	82,0
NATURE	10	64,0	66,0	77,0	81,0
POLITICS	96	**83,0**	**96,0**	81,0	**94,0**
SOCIETY	5	**80,0**	89,0	80,0	**88,0**
SPORTS	2	74,0	74,0	67,0	67,0
TERROR	18	64,0	73,0	73,0	74,0

Fonte: Autoria própria (2025).

5.1.3 Resultados quantitativos LOF

A seguir, observam-se os resultados na Tabela 14, o *Local Outlier Factor* (LOF). Este algoritmo entregou os melhores resultados dentre os três modelos não supervisionados testados. É importante destacar o mecanismo de funcionamento do LOF, que pondera cada documento e atribui um grau de *outlier*. Na prática, segundo Breunig *et al.* (2000), a densidade local é obtida a partir dos k vizinhos mais próximos. Por esse motivo, foram testadas distintas métricas de distância, sendo *cosine*, *manhattan* e *cityblock* as que entregaram os melhores resultados de

forma equitativa. Este mecanismo foi particularmente vantajoso pois não assumiu uma distribuição específica dos dados, funcionando bem mesmo em categorias com distribuição bimodal.

As categorias com melhor pontuação foram *SPORTS*, *BUSINESS*, *POLITICS* e *ARTS*, onde a acurácia obteve valores promissores, considerando a quantidade de documentos disponíveis. Para a categoria *TERROR*, este classificador interpretou de melhor forma a distribuição inerente dos documentos, alcançando quase 70% de F1 e uma acurácia de 86%, resultados superiores em relação às abordagens anteriores. As demais categorias apresentaram resultados entre 70% e 80% tanto para F1 quanto para acurácia. Além disso, o LOF demonstrou robustez em categorias com poucos eventos, como *SPORTS* (apenas 2 eventos), onde alcançou o melhor resultado individual de todo o estudo.

Finalmente, a mediana de F1 igual a 1,00 em *BUSINESS* e *POLITICS* indica que, em metade dos eventos dessas categorias, o classificador obteve classificação perfeita. Este desempenho excepcional, combinado com acurácia média superior a 79% em todas as categorias, posiciona o LOF como o método não supervisionado mais eficaz dentre as abordagens testadas.

Tabela 14 – Desempenho do LOF por categoria

Categoria	Eventos	F_1 médio (%)	F_1 mediana (%)	Acc. média (%)	Acc. mediana (%)
ACCIDENT	10	73,0	71,0	79,0	83,0
ARTS	21	**84,0**	86,0	82,0	85,0
BUSINESS	35	**83,0**	**100,0**	**87,0**	**100,0**
CRIME	10	80,0	77,0	84,0	88,0
GOVT	15	76,0	72,0	80,0	85,0
NATURE	10	75,0	80,0	**90,0**	96,0
POLITICS	96	**83,0**	**100,0**	87,0	**100,0**
SOCIETY	5	79,0	92,0	89,0	92,0
SPORTS	2	**92,0**	92,0	**95,0**	95,0
TERROR	18	69,0	73,0	86,0	90,0

Fonte: Autoria própria (2025).

5.1.4 Resultados quantitativos RAG

Embora os resultados do modelo classico usando LOF tenham sido bons, também foi testada uma abordagem mais atual. Os resultados da abordagem usando RAG são apresentados na Tabela 15. De forma geral, esta abordagem se destacou em cinco das dez categorias: *ACCIDENT* (F1 = 0,80), *BUSINESS* (F1 = 0,87), *CRIME* (F1 = 0,88), *GOVT* (F1 = 0,81) e *TERROR* (F1 = 0,71). Destaca-se particularmente o resultado em *CRIME*, onde o RAG alcançou F1 de 88%. Porém, o RAG em NATURE obteve pontuações inferiores em comparação com as outras categorias (F1 = 0,58). Este problema pode estar relacionado à presença de vocabulário técnico-científico que o *prompt* utilizado não capturou adequadamente. Por outro lado,

destaca-se a acurácia média consistentemente alta, entregando um equilíbrio adequado entre as classes.

Uma característica notável desta abordagem é a constância nos resultados, entregando valores praticamente idênticos de F1 e acurácia (média e mediana) para quase todas as categorias. Esta estabilidade sugere que um modelo de classificação que considera também características semânticas pode ser uma alternativa interessante para detecção de novidade.

Tabela 15 – Desempenho do RAG por categoria

Categoria	Eventos	F_1 médio (%)	F_1 mediana (%)	Acc. média (%)	Acc. mediana (%)
ACCIDENT	10	80,0	80,0	81,0	81,0
ARTS	21	82,0	82,0	82,0	82,0
BUSINESS	35	**87,0**	87,0	**87,0**	87,0
CRIME	10	**88,0**	88,0	**88,0**	88,0
GOVT	15	81,0	81,0	80,0	80,0
NATURE	10	58,0	58,0	60,0	60,0
POLITICS	96	82,0	82,0	82,0	82,0
SOCIETY	5	68,0	68,0	73,0	73,0
SPORTS	2	**91,0**	91,0	**91,0**	91,0
TERROR	18	71,0	73,0	72,0	72,0

Fonte: Autoria própria (2025).

5.1.5 Comparação dos resultados quantitativos entre as diferentes abordagens

Após obter os resultados das diferentes categorias para cada abordagem e classificador, foram calculadas as médias de cada métrica para comparar as abordagens deste trabalho com os *baselines* estabelecidos na literatura. A Tabela 16 demonstra que o método LOF alcançou um desempenho satisfatório de forma geral, com F1 macro de 80,90% e acurácia de 85,80%, superando o melhor resultado reportado por Ghosal *et al.* (2018) com *Random Forest* (F1 = 79,10% usando tecnicas supervisionadas). Em termos de acurácia, a abordagem LOF também obteve desempenho superior com 85,80% em comparação com a CNN de Ghosal *et al.* (2018), considerando que é uma abordagem que usa *deep learning*. Entretanto, em detecção de novidade, o LOF apresentou *recall* inferior ao da CNN e ao do próprio *Isolation Forest*.

A abordagem RAG também apresentou desempenho competitivo, com F1 de 78,84% e acurácia de 79,86%, valores próximos ao estado da arte e superiores aos *baselines* anteriores. Destaca-se que esta abordagem utiliza modelos LLM, sendo uma tecnologia mais atual e promissora para futuras investigações com diferentes *prompts* ou outras versoes de modelos LLMs. Embora o RAG seja poderoso, a qualidade do *prompt* e a especificidade do domínio influenciam fortemente no desempenho. O RAG superou os *baselines* em oito das dez categorias, com F1 médio geral de 79%. Este desempenho posiciona o RAG como uma alternativa competitiva frente as outras abordagens, com a vantagem de não requerer dados rotulados para treinamento e oferecer maior interpretabilidade através da análise contextual realizada pelo LLM.

Logo depois, o *Isolation Forest* obteve *recall* de 94,05%, o mais alto entre todas as abordagens, indicando excelente capacidade de detectar novidades, embora com precisão mais baixa (67,53%). O alto *recall* sugere que o modelo pode ser uma alternativa promissora se aplicado a outros conjuntos de dados com distribuições diferentes. No entanto, o *Elliptic Envelope* apresentou o desempenho mais limitado (F1 = 67,09%), afirmando sua sensibilidade à distribuição gaussiana presente nos dados. Como explicado nas seções anteriores, existe instabilidade inerente à distribuição dos dados uados, o que afetou particularmente esses dois classificadores.

Comparando os resultados com os *baselines* de Ghosal *et al.* (2018) (F1 = 79%) e Nair (2023) (F1 = 72% usando aprendizado não supervisionadas), as abordagens propostas, especialmente LOF e RAG, demonstram resultados competitivos ou superiores. O melhor *baseline* identificado foi o RDV-CNN de Ghosal *et al.* (2018), que alcançou F1 macro de 83,50%, acurácia de 84,53% e F1(N) de 86%, representando o estado da arte no *corpus* TAP-DLND-1.0.

Tabela 16 – Comparação das abordagens de detecção de novidade no *corpus* TAP-DLND 1.0.

Sistema	**Train/Test**	**$\mathbf{F_1}$ macro**	**Acc.**	**P(N)**	**R(N)**	**$\mathbf{F_1}$(N)**
Baseline 1: Paragraph Vector+LR	10-fold CV	72,00%	72,81%	75,00%	75,00%	75,00%
Baseline 2: BiLSTM+MLP	10-fold CV	77,00%	78,57%	78,00%	84,00%	80,00%
Set Difference+LR (Zhang 2002)	10-fold CV	72,50%	73,21%	74,00%	71,00%	72,00%
Geometric Distance+LR (Zhang 2002)	10-fold CV	69,50%	69,84%	65,00%	84,00%	73,00%
LM:Dirichlet Prior+LR (Zhang 2002)	10-fold CV	73,50%	73,62%	73,00%	74,00%	74,00%
Novelty (IDF)+LR (Karkali 2013)	10-fold CV	45,50%	54,26%	52,00%	92,00%	66,00%
Supervised features Ghosal *et al.* (2018)	10-fold CV	78,50%	79,27%	77,00%	82,00%	79,00%
RDV-CNN Ghosal *et al.* (2018)	10-fold CV	**83,50%**	84,53%	**86,00%**	87,00%	**86,00%**
Aprendizado Não Supervisionado Nair (2023)	por tópico	72,00%	72,00%	70,50%	75,00%	72,50%
Abordagens propostas neste trabalho (por evento)						
TF-IDF + LOF	por evento	80,90%	85,80%	82,00%	83,00%	81,00%
TF-IDF + Isolation Forest	por evento	69,33%	75,97%	67,53%	**94,05%**	74,95%
TF-IDF + Elliptic Envelope	por evento	67,09%	61,80%	58,25%	93,57%	67,09%
RAG (DeepSeek Embeddings + BGE)	por evento	78,84%	79,86%	81,29%	77,06%	77,93%

Fonte: Autoria própria (2025). Resultados dos *baselines* extraídos de Ghosal *et al.* (2018) e Nair (2023).

É importante observar que as abordagens propostas neste trabalho utilizam avaliação por evento, enquanto os *baselines* empregam validação cruzada 10-fold. Apesar dessa diferença metodológica, os resultados indicam que as técnicas propostas, especialmente LOF e RAG, representam avanços significativos para a tarefa de detecção de novidade. Adicionalmente, destaca-se que as pesquisas anteriores não apresentam resultados detalhados por evento ou categoria, dificultando uma análise aprofundada. Essa falta de informação específica é problemática, considerando que métricas como acurácia e F1 dependem fortemente do contexto e da distribuição dos dados em cada categoria, tornando essencial compreender o comportamento dos modelos em diferentes cenários.

Um excelente exemplo deste problema pode ser observado ao comparar a abordagem LOF com o RAG. Os resultados do LOF foram consistentes para quase todas as categorias,

com valores entre 70% e 92%, mantendo-se predominantemente na faixa de 70% a 80%, o que demonstra estabilidade e previsibilidade nos resultados. Por outro lado, o RAG apresentou duas categorias com comportamento atípico: *NATURE* com 58% e *SPORTS* com 91%, valores que destoam do padrão geral de 70% a 80% observado nas demais categorias. Essa variabilidade não fica evidente quando se considera apenas a média geral. Assim, a comparação por categorias auxilia na interpretação dos dados e permite escolher de forma mais fundamentada o classificador adequado para cada contexto.

6 CONCLUSÃO

Este trabalho propôs varios modelos não supervisionados para detecção de novidades baseados em técnicas clássicas e técnicas atuais, testados e apresentados. Para alcançar esses sistemas, foi feito um levantamento bibliográfico sobre métodos e algoritmos não supervisionados empregados na detecção de novidade. Em seguida, o estado da arte foi levantado, revelando que a detecção de novidade pode ser tratada como um problema de detecção de anomalias. Complementarmente, pesquisas mais atuais revelaram que métodos mais modernos também podem ser uma solução ao problema, neste caso, RAG com o LLM DeepSeek.

Para desenvolver a metodologia proposta, foi identificado um conjunto de dados relevante baseado em *benchmarks* atuais. Este conjunto contém documentos de novidade, particularmete, notícias de diferentes categorias. Também, foram apresentados diferentes mecanismos para avaliar os resultados de maneira quantitativa, comparando-os com *baselines* da literatura.

Adicionalmente, foram produzidos distintas abordagens: a primeira utilizando diferentes classificadores classicos (LOF, *Isolation Forest* e *Elliptic Envelope*), e o segundo baseado na arquitetura RAG. Em ambas as abordagens, o sistema não tinha conhecimento dos rótulos do conjunto de dados *target*, focando unicamente nas características léxicas e semânticas do texto. Os dois melhores resultados foram obtidos pelo classificador LOF e pela arquitetura RAG. O RAG apresentou valores de acurácia e F1-score superiores a 70%, enquanto o LOF alcançou acurácia e F1-score superiores a 80%, valores bastante razoáveis se comparados com os encontrados na literatura. Considerando a complexidade inerente à abordagem não supervisionada, o LOF chegou a superar o estado da arte para este conjunto de dados em termos de acurácia.

Conclui-se que as estratégias propostas são eficazes e representam alternativas escaláveis para a detecção de novidades em textos. Os sistemas LOF e RAG desenvolvidos demonstraram desempenho comparável, e em alguns casos superior, a técnicas que utilizam validação cruzada *10-fold*. Considerando um treinamento realizado para cada conjunto de eventos, a solução proposta se aproxima mais de um contexto real, onde as notícias devem ser avaliadas de acordo com o tópico específico.

Por outro lado, os autores dos trabalhos relacionados que utilizam o mesmo conjunto de dados não apresentam resultados por categoria, o que dificulta a avaliação do desempenho dos classificadores em cenários específicos. Para avaliar adequadamente os resultados de detecção de novidade, torna-se necessária uma comparação que considere características como a distribuição e a quantidade de eventos em cada categoria. Conforme demonstrado nos resultados deste trabalho, essas características dos dados impactam diretamente no desempenho dos modelos, sendo fundamental considerar tais variações na escolha e avaliação de técnicas de detecção de novidade.

Sugestões para continuidade ou aprimoramento deste trabalho acadêmico são listadas a seguir:

- Utilização de outros bancos de dados em domínios como educação, análise de tendências e monitoramento de informação.
- Ampliar o conjunto de dados e anotações para línguas diferentes, como português ou espanhol, considerando a falta de conjuntos de dados de *benchmark* nesses idiomas.
- Estudar outros métodos mais atuais ou diferentes estratégias de prompts para aprimorar a abordagem que utiliza LLMs.

Por fim, pode-se concluir que os objetivos propostos foram atingidos. As abordagens desenvolvidas mostram-se interessantes para aplicação em áreas como jornalismo automatizado, educação, análise de tendências e monitoramento de informação, sobretudo quando é possível realizar a divisão por tópicos. Espera-se que este trabalho não apenas apresente uma análise quantitativa dos resultados, mas também contribua para um entendimento mais amplo da tarefa de detecção de novidades. A expectativa é que o processo desenvolvido sirva como base para pesquisas futuras e melhorias contínuas.

REFERÊNCIAS

AGGARWAL, C. C. **Outlier Analysis**. 2. ed. Cham: Springer, 2017.

AHMED, M.; MAHMOOD, A. N.; HU, J. A survey of network anomaly detection techniques. **J. Netw. Comput. Appl.**, Academic Press Ltd., GBR, v. 60, n. C, p. 19–31, jan. 2016. ISSN 1084-8045. Disponível em: https://doi.org/10.1016/j.jnca.2015.11.016.

AI, D. **DeepSeek Large Language Model Documentation**. 2024. https://www.deepseek.com/. Acesso em: 10 fev. 2025.

AL., H. et. **Array programming with NumPy**. 2020.

BELCIC, I.; STRYKER, C. **RAG vs. fine-tuning**. 2024. IBM Think. Acesso em: 14 nov. 2025. Disponível em: https://www.ibm.com/think/topics/rag-vs-fine-tuning.

BISHOP, C. M. **Pattern Recognition and Machine Learning**. New York: Springer, 2006.

BOMMASANI, R. *et al.* **On the Opportunities and Risks of Foundation Models**. 2022. Disponível em: https://arxiv.org/abs/2108.07258.

BREUNIG, M. M. *et al.* LOF: Identifying density-based local outliers. **SIGMOD Record**, v. 29, n. 2, p. 93–104, 2000.

CHALAPATHY, R.; CHAWLA, S. **Deep Learning for Anomaly Detection: A Survey**. 2019. Disponível em: https://arxiv.org/abs/1901.03407.

CHANDOLA, V.; BANERJEE, A.; KUMAR, V. Anomaly detection: A survey. **ACM Computing Surveys**, v. 41, n. 3, p. 1–58, 2009.

DASGUPTA, T.; DEY, L. Automatic scoring for innovativeness of textual ideas. *In*: **The Workshops of the Thirtieth AAAI Conference on Artificial Intelligence**. [*S.l.*]: AAAI Press, 2016. (Technical Report WS-16-10), p. 507–511.

DEVLIN, J. *et al.* Bert: Pre-training of deep bidirectional transformers for language understanding. **arXiv preprint arXiv:1810.04805**, 2018.

FOUNDATION, P. S. **Python Language Reference**. 2024. Https://www.python.org.

GARCÍA, S.; HERRERA, F. Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy. **Evol. Comput.**, MIT Press, Cambridge, MA, USA, v. 17, n. 3, p. 275–306, set. 2009. ISSN 1063-6560. Disponível em: https://doi.org/10.1162/evco.2009.17.3.275.

GHOSAL, T. *et al.* Novelty goes deep. a deep neural solution to document level novelty detection. *In*: BENDER, E. M.; DERCZYNSKI, L.; ISABELLE, P. (Ed.). **Proceedings of the 27th International Conference on Computational Linguistics**. Santa Fe, New Mexico, USA: Association for Computational Linguistics, 2018. p. 2802–2813. Acesso em: 14 nov. 2025. Disponível em: https://aclanthology.org/C18-1237/.

GHOSAL, T. *et al.* Is your document novel? let attention guide you. an attention-based model for document-level novelty detection. **Natural Language Engineering**, v. 27, p. 427 – 454, 2020. Disponível em: https://api.semanticscholar.org/CorpusID:219029801.

GHOSAL, T. *et al.* TAP-DLND 1.0: A corpus for document level novelty detection. *In*: **Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)**. Miyazaki, Japan: European Language Resources Association (ELRA), 2018. Acesso em: 8 set. 2025. Disponível em: https://aclanthology.org/L18-1559/.

GOOGLE. **Google Cloud – Large Language Models (LLMs)**. 2025. https://cloud.google.com/ai/llms?hl=es. Acesso em: 14 fev. 2025.

GRUS, J. **Data Science from Scratch: First Principles with Python**. 2. ed. Sebastopol: O'Reilly Media, 2019.

GéRON, A. **Mãos à Obra: Aprendizado de Máquina com Scikit-Learn, Keras & TensorFlow: Conceitos, Ferramentas e Técnicas para a Construção de Sistemas Inteligentes**. 2. ed. Rio de Janeiro: Alta Books, 2021.

HAWKINS, D. M. **Identification of Outliers**. London: Chapman and Hall, 1980.

Hugging Face. **Hugging Face: The AI Community Building the Future**. 2023. Acesso em: 18 nov. 2025. Disponível em: https://huggingface.co.

HUNTER, J. D. **Matplotlib: A 2D Graphics Environment**. 2007.

JURAFSKY, D.; MARTIN, J. H. **Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition**. 2020. Acesso em: 8 set. 2025. Disponível em: https://web.stanford.edu/~jurafsky/slp3/.

KARKALI, M. *et al.* Efficient online novelty detection in news streams. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 8180 LNCS, n. PART 1, p. 57–71, 2013.

LE, Q. V.; MIKOLOV, T. **Distributed Representations of Sentences and Documents**. 2014. Disponível em: https://arxiv.org/abs/1405.4053.

LIU, F. T.; TING, K. M.; ZHOU, Z.-H. Isolation forest. *In*: **2008 Eighth IEEE International Conference on Data Mining**. [*S.l.*: *s.n.*], 2008. p. 413–422.

LIU, Y. *et al.* **Harnessing Large Language Models for Scientific Novelty Detection**. 2025. Disponível em: https://arxiv.org/abs/2505.24615.

LOGUNOVA, O. B. I. **Word2Vec: Why Do We Need Word Representations?** 2023. https://serokell.io/blog/word2vec. Imagem extraída de artigo publicado em 08 maio 2023.

MANNING, C. D.; RAGHAVAN, P.; SCHüTZE, H. **Introduction to Information Retrieval**. Cambridge: Cambridge University Press, 2008.

MCENERY, T.; HARDIE, A. **Corpus Linguistics: Method, Theory and Practice**. Cambridge: Cambridge University Press, 2012.

MIKOLOV, T. *et al.* **Efficient Estimation of Word Representations in Vector Space**. 2013. Disponível em: https://arxiv.org/abs/1301.3781.

NAIR, B. Predicting document novelty: an unsupervised learning approach. **Knowledge and Information Systems**, v. 66, p. 1709 – 1728, 2023. Disponível em: https://api.semanticscholar.org/CorpusID:264053054.

OPENAI. **Generative Pretrained Transformer**. 2023. Available online: https://openai.com.

POWERS, D. M. W. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation. **Journal of Machine Learning Technologies**, v. 2, n. 1, p. 37–63, 2011. Acesso em: 8 set. 2025. Disponível em: https://bioinfopublication.org/files/articles/2_1_1_JMLT.pdf.

SAMUEL, A. L. Some studies in machine learning using the game of checkers. **IBM Journal of Research and Development**, v. 3, n. 3, p. 210–229, 1959.

Scikit-Learn Developers. **Scikit-learn: Machine Learning in Python**. 2024. Acesso em: 8 set. 2025. Disponível em: https://scikit-learn.org/stable/.

TAHA, A. A.; HANBURY, A. Metrics for evaluating 3D medical image segmentation: Analysis, selection, and tool. **BMC Medical Imaging**, v. 15, p. 29, 2015.

TEAM, T. pandas development. **pandas-dev/pandas: Pandas**. 2024. Https://pandas.pydata.org/.

TENG, H. S.; CHEN, K.; LU, S. C.-Y. Adaptive real-time anomaly detection using inductively generated sequential patterns. *In*: **Proceedings of the 1990 IEEE Computer Society Symposium on Research in Security and Privacy**. Los Alamitos: IEEE, 1990. p. 278–284.

TUNSTALL, L.; WERRA, L. von; WOLF, T. **Natural Language Processing with Transformers: Building Language Applications with Hugging Face**. O'Reilly Media, 2022. ISBN 9781098103248. Disponível em: https://books.google.com.br/books?id=pNBpzwEACAAJ.

VASWANI, A. *et al.* **Attention Is All You Need**. 2023. Disponível em: https://arxiv.org/abs/1706.03762.

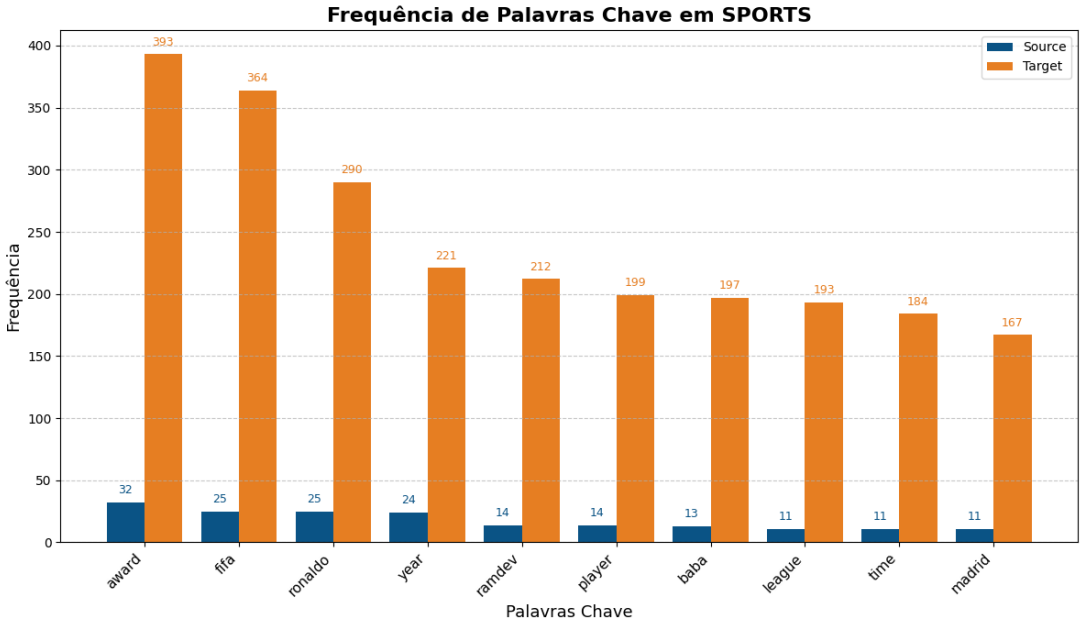
VERLEYSEN, M.; FRANÇOIS, D. The curse of dimensionality in data mining and time series prediction. *In*: CABESTANY, J.; PRIETO, A.; SANDOVAL, F. (Ed.). **Computational Intelligence and Bioinspired Systems**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. p. 758–770.

WASKOM, M. **Seaborn: statistical data visualization**. 2021. Https://seaborn.pydata.org/.

ZHANG, Y.; CALLAN, J.; MINKA, T. Novelty and redundancy detection in adaptive filtering. *In*: **Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval**. New York, NY, USA: Association for Computing Machinery, 2002. (SIGIR '02), p. 81–88. ISBN 1581135610. Disponível em: https://doi.org/10.1145/564376.564393.

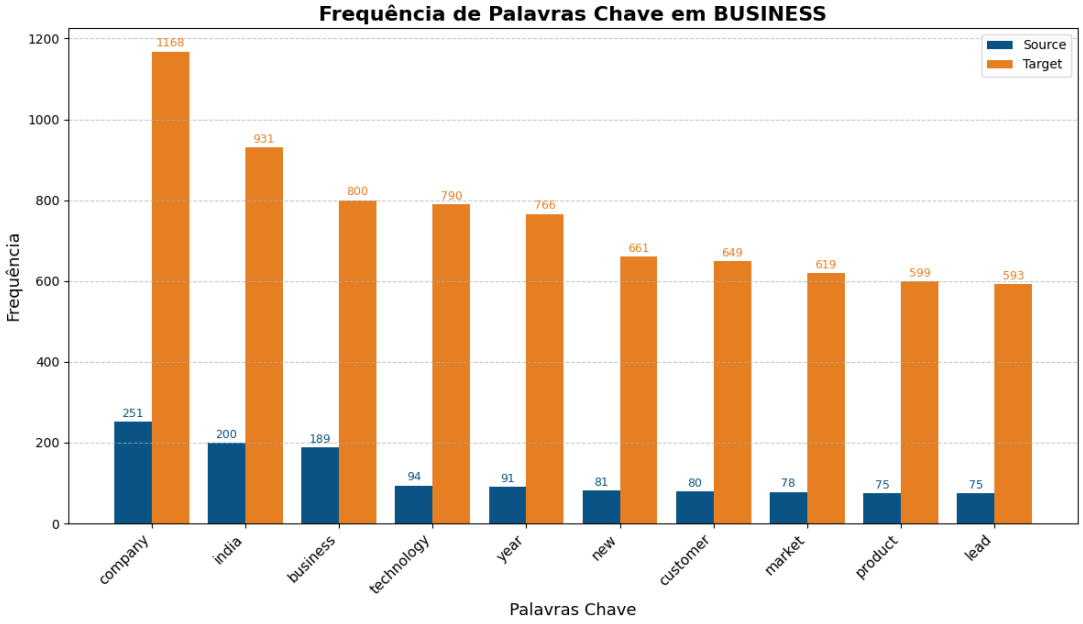
APÊNDICE A – Distribuição de Frequência de Palavras-chave na categoria e Entidades Nomeadas por Categoria

Figura 20 – Distribuição de palavras chave na categoria SPORTS.



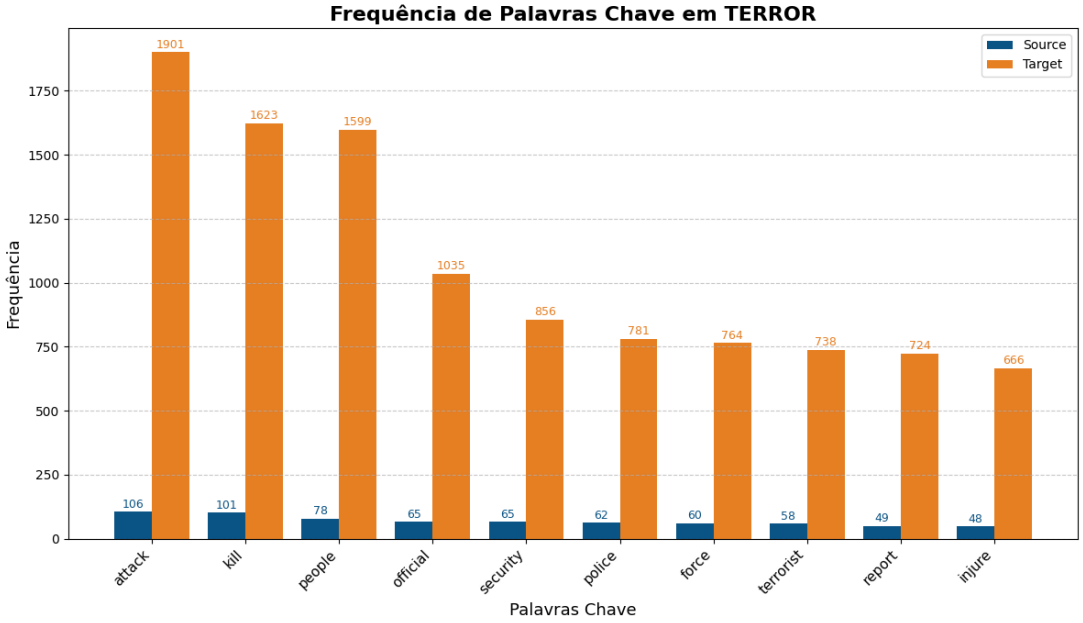
Fonte: Autoria própria (2025).

Figura 21 – Distribuição de palavras chave na categoria BUSINESS.



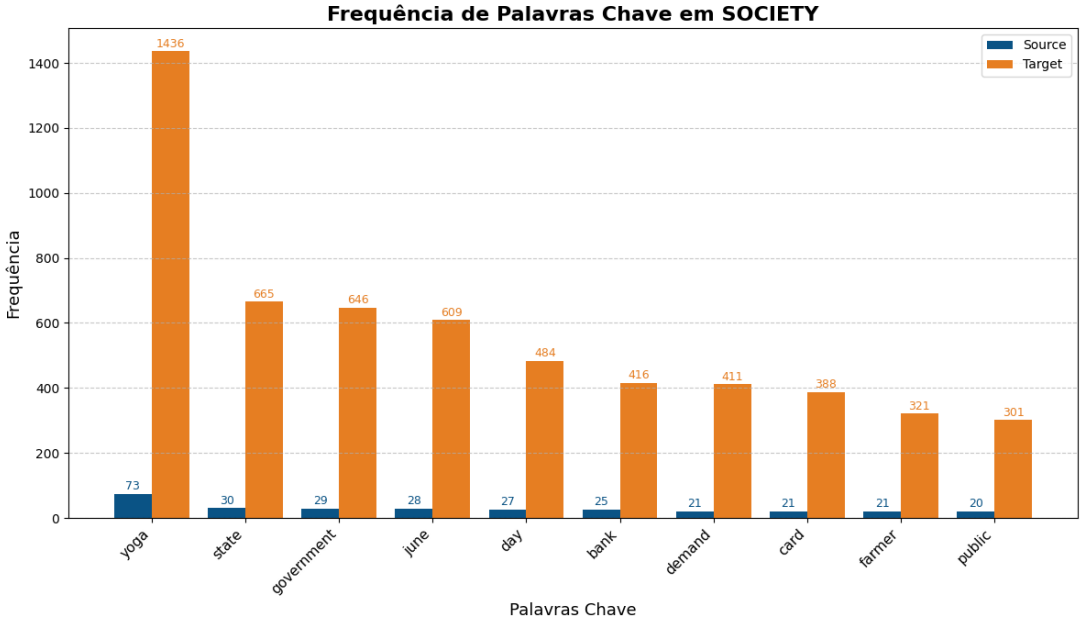
Fonte: Autoria própria (2025).

Figura 22 – Distribuição de palavras chave na categoria TERROR.



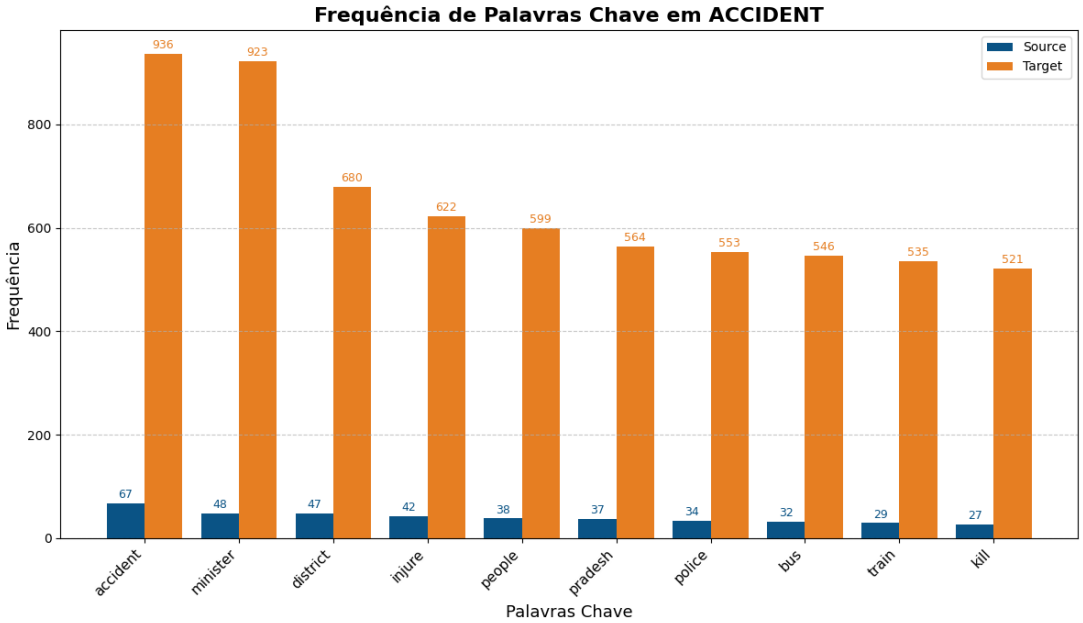
Fonte: Autoria própria (2025).

Figura 23 – Distribuição de palavras chave na categoria SOCIETY.



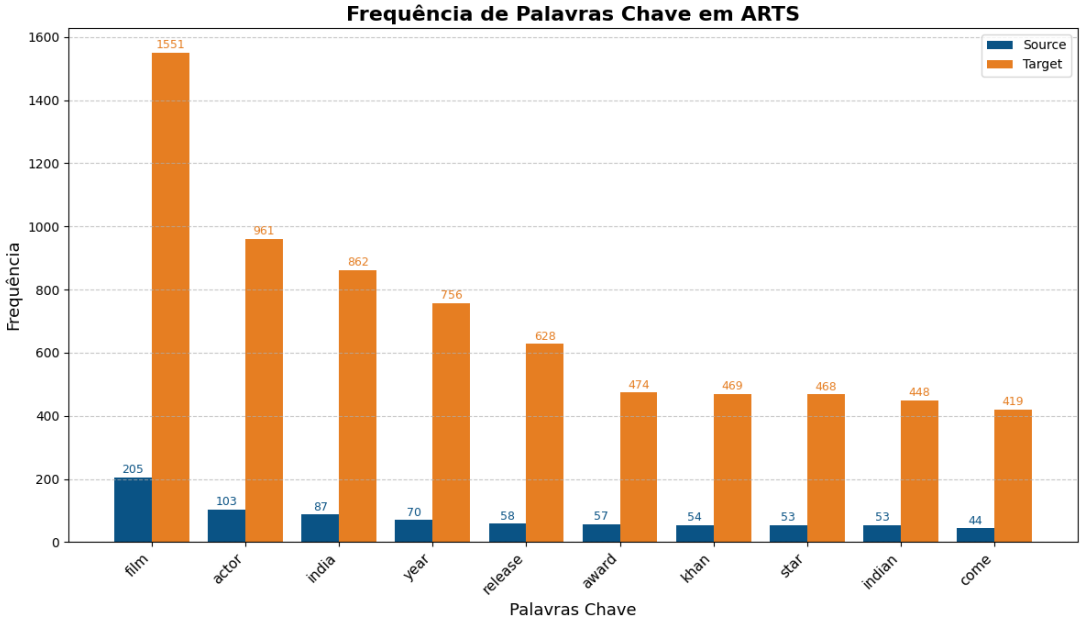
Fonte: Autoria própria (2025).

Figura 24 – Distribuição de palavras chave na categoria ACCIDENT.



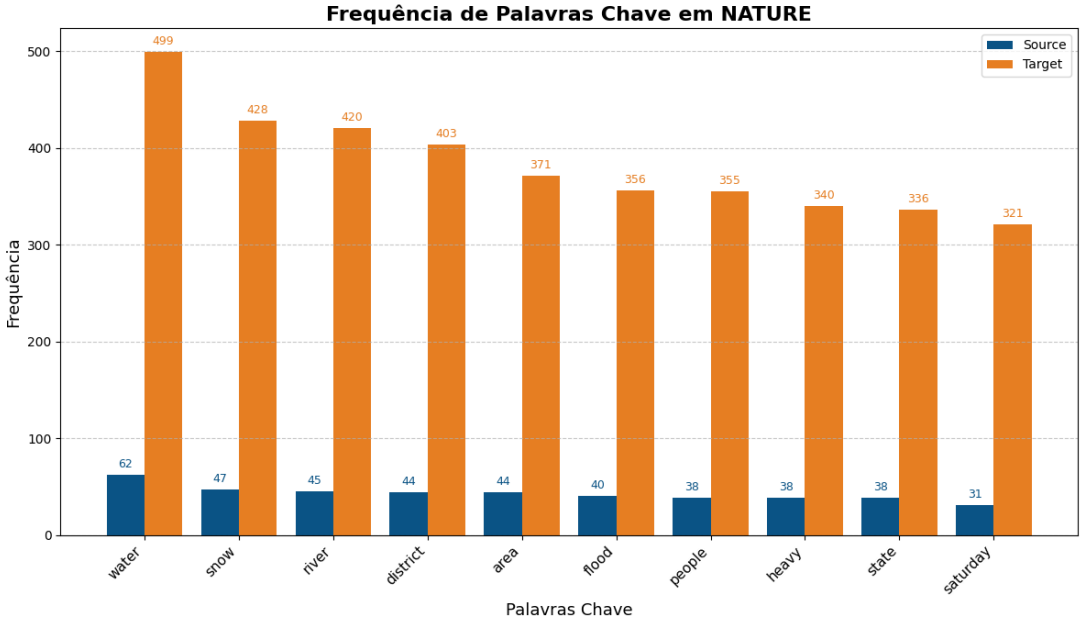
Fonte: Autoria própria (2025).

Figura 25 – Distribuição de palavras chave na categoria ARTS.



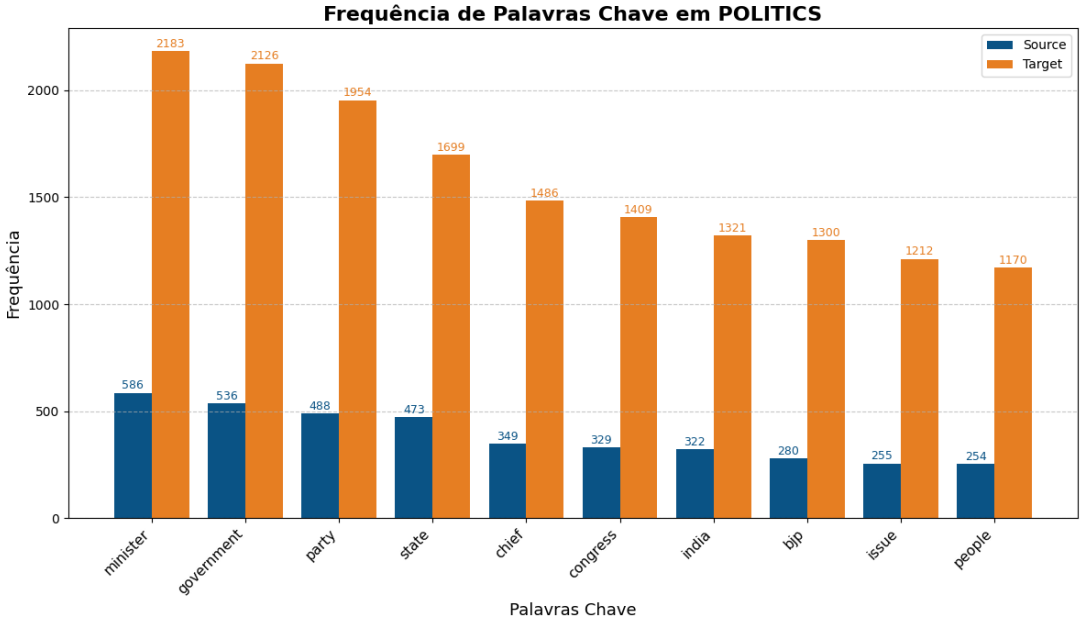
Fonte: Autoria própria (2025).

Figura 26 – Distribuição de palavras chave na categoria NATURE.



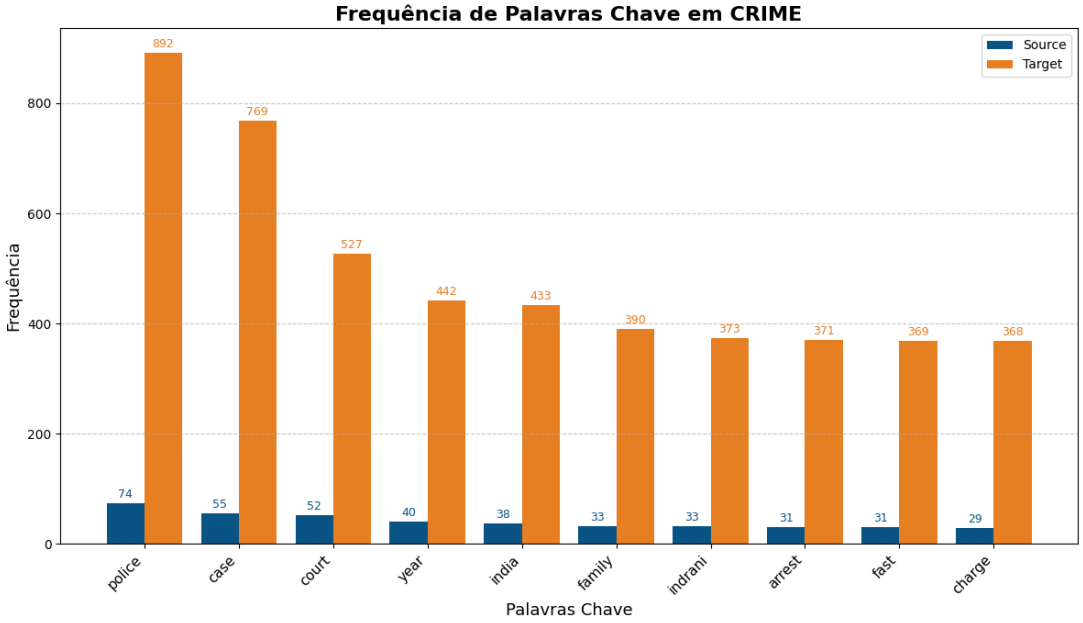
Fonte: Autoria própria (2025).

Figura 27 – Distribuição de palavras chave na categoria POLITICS.



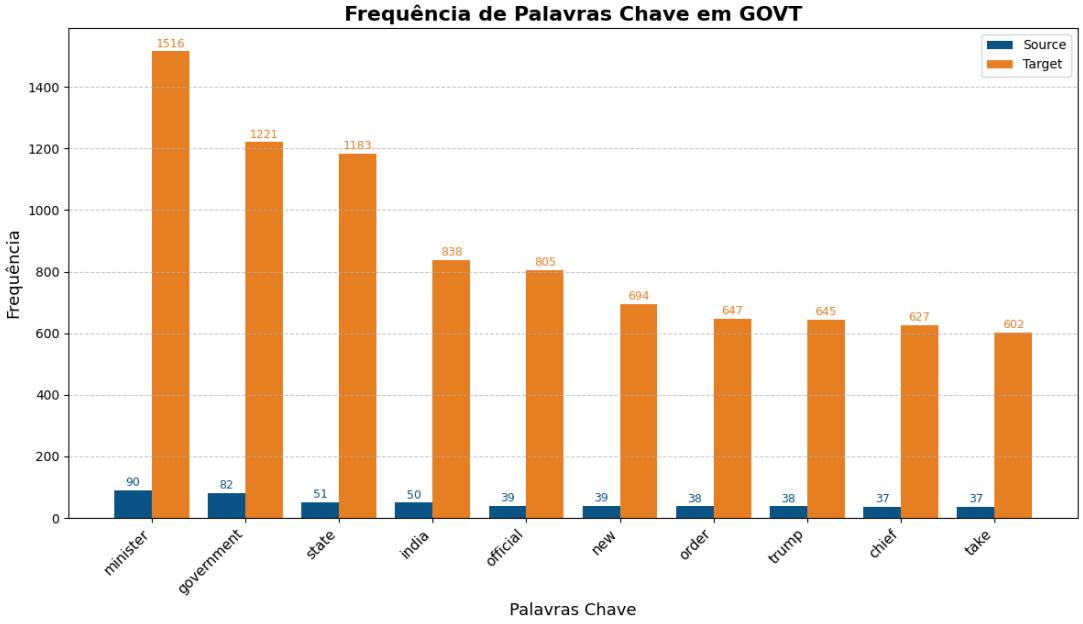
Fonte: Autoria própria (2025).

Figura 28 – Distribuição de palavras chave na categoria CRIME.



Fonte: Autoria própria (2025).

Figura 29 – Distribuição de palavras chave na categoria GOVT.



Fonte: Autoria própria (2025).

ANEXO A – Exemplo de Instância do Corpus TAP-DLND (Categoria SPORTS)

Este anexo apresenta um exemplo de notícia real extraída do corpus TAP-DLND , utilizado nos experimentos de detecção de novidade (GHOSAL *et al.*, 2018). O trecho a seguir pertence à categoria `SPORTS` e representa uma instância do conjunto *source* (notícia base).

Título da notícia

Baba Ramdev Calls Out Olympic Wrestling Medallist And Challenges Him To A 'Dangal'!

Texto completo

Even as the Pro Wrestling League continues, what draws the most interest is a match featuring none other than Baba Ramdev himself! Yes, it's true. The man has challenged 2008 Olympic silver medallist Andrey Stadnik for a friendly 'dangal'. The Ukrainian said he was surprised with the offer but perhaps didn't have much of a choice but to accept the challenge. Well, it will be interesting to see what happens. One must remember that Baba Ramdev is an extremely fit man.

Metadados

- **Categoria:** SPORTS
- **Fonte:** Indiatimes.com