UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Aprendizado de máquina baseado em Classificação hierárquica de textos aplicado ao contexto chamados de suporte

Beatriz Yokota

Monografia - MBA em Inteligência Artificial e Big Data

ICMC USP SÃO CARLOS

Beatriz Yokota

Aprendizado de máquina baseado em Classificação hierárquica de textos aplicado ao contexto chamados de suporte

Monografia apresentada ao Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, como parte dos requisitos para obtenção do título de Especialista em Inteligência Artificial e Big Data.

Área de concentração: Inteligência Artificial

Orientador: Prof. Dr. Tiago A. Almeida

Co-orientador: Dr. Brucce Neves dos Santos

Versão original

São Carlos

2025

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi, ICMC/USP, com os dados fornecidos pelo(a) autor(a)

S856m

Yokota, Beatriz
Aprendizado de máquina baseado em Classificação hierárquica de textos aplicado ao contexto chamados de suporte / Beatriz Yokota ; orientador Tiago A. Almeida ; co-orientador Dr. Brucce Neves dos Santos. – São Carlos, 2025.
77 p. : il. (algumas color.) ; 30 cm.

Monografia (MBA em Inteligência Artificial e Big Data) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2025.

1. NLP. 2. Classificação Hierárquica. 3. LCL. I. Almeida, Tiago A, orient. II. Santos, Brucce Neves dos Santos, co-orient.

Beatriz Yokota

Aprendizado de máquina baseado em Classificação hierárquica de textos aplicado ao contexto chamados de suporte

Monograph presented to the Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, as part of the requirements for obtaining the title of Specialist in Artificial Intelligence and Big Data.

Concentration area: Artificial Intelligence

Advisor: Prof. Dr. Tiago A. Almeida
Coadvisor: Dr. Brucce Neves dos Santos

Original version

São Carlos

2025

RESUMO

Yokota, B. **Aprendizado de máquina baseado em Classificação hierárquica de textos aplicado ao contexto chamados de suporte**. 2025. 77p. Monografia (MBA em Inteligência Artificial e Big Data) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2025.

Este trabalho apresenta um método de Classificação Hierárquica (CH) de textos para automatização da categorização de chamados de suporte. O estudo é motivado pelos desafios encontrados no cenário atual de atendimento ao cliente: o alto volume de chamados diários e a complexidade da estrutura taxonômica hierárquica das categorias. A metodologia proposta emprega um classificador base e utiliza a abordagem *Local Classifier per Level* (LCL), na qual para cada nó não-folha é criado um algoritmo classificador e os rótulos são seus nós filhos. Com isso, mantém a consistência das predições através dos diferentes níveis hierárquicos, oferecendo vantagens práticas como a redução do número de classificadores necessários e maior facilidade de extensão para novos domínios. No presente estudo, adotou-se *Naive Bayes* (NB) e *Multilayer Perceptron* (MLP) com duas variações arquiteturais: MLP(256, 128) e MLP(512) como classificadores base, e o desempenho da metodologia foi avaliado por meio da métrica $F1_{macro}$, levando em consideração o significativo desbalanceamento entre as classes na base de dados. Os resultados obtidos pelo método proposto revelaram desafios substanciais na classificação automática de textos especializados, evidenciando a necessidade de abordagens metodológicas mais sofisticadas para capturar nuances semânticas em taxonomias complexas.

Palavras-chave: NLP. Classificação Hierárquica. LCL.

ABSTRACT

Yokota, B. **Machine learning based on Text Hierarchy Classification applied to the context called support**. 2025. 77p. Monograph (MBA in Artificial Intelligence and Big Data) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2025.

This study presents a Hierarchical Classification (HC) method for text-based automation of support ticket categorization. The research is motivated by challenges in the current customer service scenario: the high daily volume of tickets and the complexity of the hierarchical taxonomic structure of categories. The proposed methodology employs a base classifier and utilizes the Local Classifier per Level (LCL) approach, where a classification algorithm is created for each non-leaf node, with its child nodes as labels. This approach maintains prediction consistency across different hierarchical levels, offering practical advantages such as reducing the number of required classifiers and facilitating extension to new domains. In the present study, Naive Bayes (NB) and Multilayer Perceptron (MLP) was adopted with two architectural variations: MLP(256, 128) and MLP(512) as base classifiers, and the methodology's performance was evaluated using the $F1_{macro}$ metric, considering the significant class imbalance in the database. The results obtained by the proposed method revealed substantial challenges in the automatic classification of specialized texts, highlighting the need for more sophisticated methodological approaches to capture semantic nuances in complex taxonomies.

Keywords: NLP. Hierarchical Classification. LCL.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AM	Aprendizado de Máquina
AUC	*Area Under ROC Curve*
BERT	*Bidirectional Encoder Representations from Transformers*
BCE	*Binary Cross-Entropy*
BoW	*Bag-of-Words*
CART	*Classification and Regression Trees*
CHAMP	*Comprehensive Hierarchy Aware Multi-label Predictions*
Clus-HMC	*Hierarchical multi-label classification with Predictive Clustering Trees*
CNNs	*Convolutional Neural Networks*
CSHCIC	*Cost-Sensitive Hierarchical Classification with Imbalanced Classes*
CT	Categorização de Textos
DAG	*Direct Acyclic Graphs*
DSM	*Distributional Semantic Models*
DT	*Decision Trees*
ELMo	*Embeddings from Language Models*
FAQs	Perguntas Mais Frequentes
FK	*Filter Kernels*
FP	Falsos Positivos
FMA	*Free Music Archive*
FN	Falsos Negativos
GC	*Global Classifier*
GMNB	*Global Model Naive Bayes*
GPT	*Generative Pre-trained Transformer*
GVNS ou GVNS-FSHC	*General Variable Neighborhood Search*

HMC-LMLP *Hierarchical Multi-Label Classification with Local Multi-Layer Perceptrons*

HPT *Hierarchy-aware Prompt Tuning*

HSIM *Hierarchical Supervised Imputation Method*

HSVM *Hierarchical-SVM*

HSVM-S *Global Margin Maximization*

IA Inteligência Artificial

IDC *International Data Corporation*

IDF *Inverse Document Frequency*

KNN *K-nearest neighbours*

LCL *Local Classifier per Level*

LCN *Local Classifier per Node*

LCPN *Local Classifier per Parent Node*

MLNP *Mandatory Leaf-Node Prediction*

MLP *Multilayer Perceptron*

MSD Modelos Semânticos Distribucionais

NB *Naive Bayes*

NMLNP *Non-Mandatory Leaf Node Prediction*

ODP *Open Directory Project's*

PLN Processamento de Linguagem Natural

PMI *Pointwise Mutual Information*

RF *Random Forest*

RNAs Redes Neurais Artificiais

ROC *Receiver Operating Characteristic Curve*

SC *Structured Classification*

SVM *Support Vector Machines*

TF	*Term Frequency*
TF-IDF	*Term Frequency-Inverse Document Frequency*
TFN	Taxa de Falsos Negativos
TFP	Taxa de Falsos Positivos
TVP	Taxa de Verdadeiros Positivos
VNS	*Variable Neighborhood Search*
VP	Verdadeiros Positivos
VN	Verdadeiros Negativos
ZB	Zettabytes

SUMÁRIO

1 INTRODUÇÃO

Com o avanço da tecnologia e a aceleração do processo de digitalização, uma enorme quantidade de informações processadas diariamente por corporações passou a ser armazenada em formato não-estruturado, como por exemplo: e-mails, memorandos, notas de *call centers* e operações de suporte, notícias, grupos de usuários, chats, relatórios, cartas, pesquisas, *white papers*, material de marketing, pesquisas, apresentações e páginas da Web (BLUMBERG; ATRE, 2003). A *International Data Corporation* (IDC) estimou que os dados globais crescerão de 33 Zettabytes (ZB) em 2018 para 175 ZB em 2025 (REINSEL; GANTZ; RYDNING, 2018). Contudo, os dados só terão relevância se forem devidamente categorizados e examinados (GANTZ; REINSEL, 2012). Apesar da enorme quantidade de dados gerados diariamente, poucos são utilizados para extrair relatórios, identificar eventos e tendências, apoiar na tomada de decisão com base em análises e inferências estatísticas para fins comerciais, operacionais ou científicos.

Uma aplicação envolvendo dados não-estruturados que existe no ambiente corporativo é a classificação de chamados. Muitas organizações possuem centrais de atendimento para suporte/assistência em forma de chat, na qual o cliente interage com pessoas (ou *chatbots* automáticos) por meio de trocas de mensagens. As informações geradas através desta interação (histórico da conversa e/ou campos preenchidos pelo atendente) podem ser organizadas por temas/assuntos, sendo que cada tópico representa uma categoria distinta, conforme ilustrado na Tabela 1.

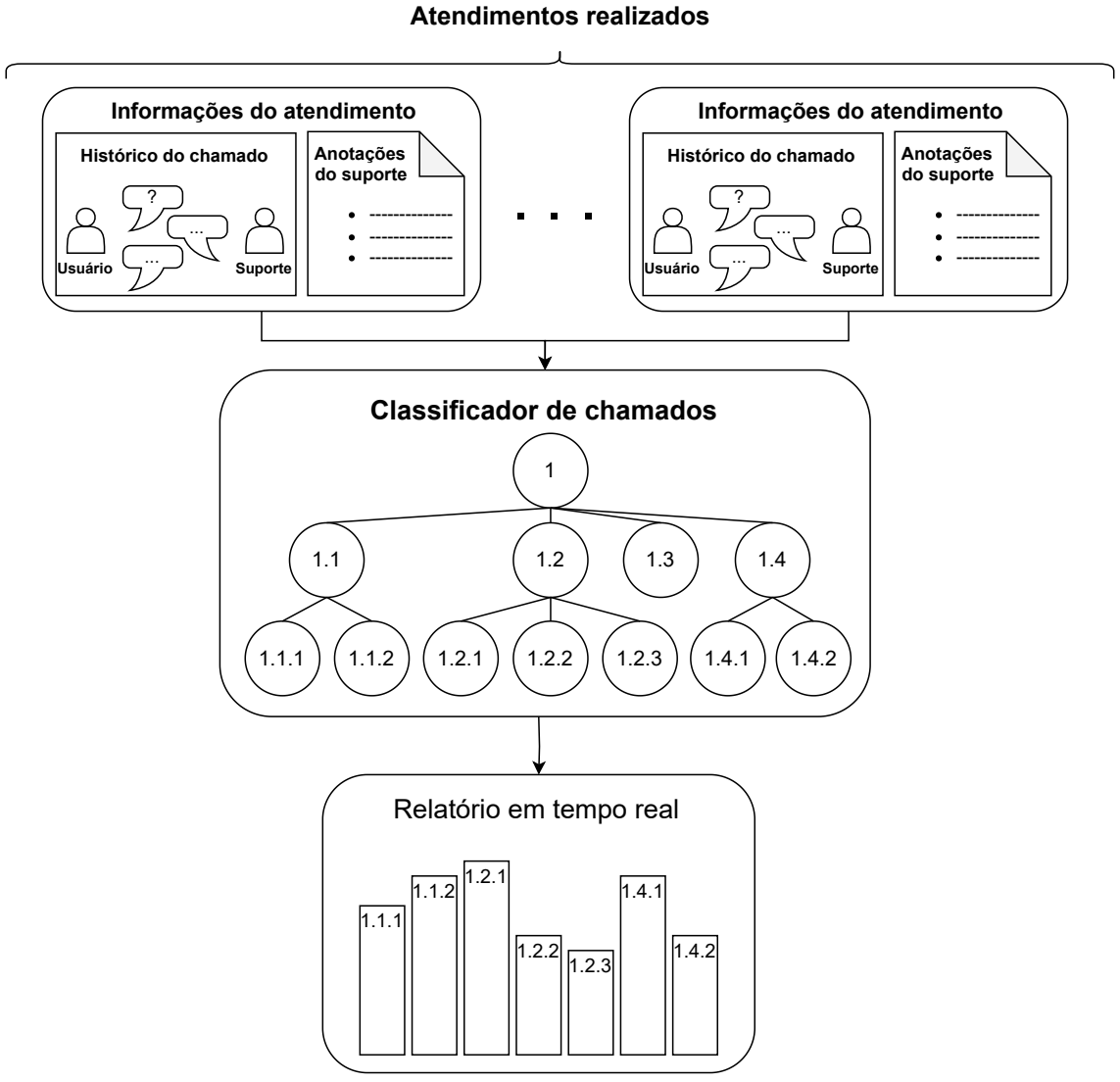
Tabela 1 – Exemplos temas de chamados

Categoria	**Subcategoria**
Emissão de NF-e	Como emitir NF-e Falha na emissão de NF-e Cancelamento e inutilização de NF-e
Cadastro de compras	Cadastro manual Cadastro via importação de Nota Fiscal
Cadastro de clientes	Cadastro completo Cadastro rápido
Cadastro de vendas	Frente de caixa *E-commerce*
Controle de estoque	Cadastro manual Cadastro via importação de Nota Fiscal

A possibilidade de rotular os assuntos dos chamados automaticamente permite tomar decisões de negócio e acionar outras áreas quando há problemas urgentes, como,

por exemplo, bugs e instabilidades, conforme ilustrado na Figura 1. Contudo, além do grande volume de atendimentos realizados diariamente, analisar todas as informações dos chamados manualmente e, em tempo real, torna-se inviável. Isso também se deve ao fato de que transformar dados em informações úteis é um processo lento, caro e subjetivo dependendo de quem está rotulando (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996). Além disso, os temas/assuntos dos chamados podem ser organizados em uma taxonomia, havendo categorias e subcategorias que se relacionam de maneira hierárquica.

Figura 1 – Exemplo de processo de classificação de chamados



Fonte: Autor

Diante deste cenário, a categorização hierárquica de textos, também conhecida como Classificação Hierárquica (CH) de textos, tornou-se promissora (SEBASTIANI, 2001; NIGAM, 2001). Por se tratar de uma metodologia de Aprendizado de Máquina (AM), a CH de textos possibilita a automatização do processo de rotulagem, tornando-o mais eficiente para as organizações (NAIK; RANGWALA, 2018). Ao construir um modelo de AM que classifique os chamados, seria possível utilizá-lo em outras aplicações voltadas

ao atendimento do consumidor, como: usar como insumo para treinamentos de *chatbots* automáticos para identificação mais rápida do assunto e inserção dos assuntos em sistemas de Perguntas Mais Frequentes (*Frequently Asked Questions* - FAQs) bem como portais de perguntas e respostas.

A CH de textos se despontou como a abordagem mais promissora e apropriada, pois esse tipo de classificação considera as inter-relações entre as classes e possibilita organizar os documentos em uma estrutura hierárquica (MENG *et al.*, 2019). Além disso, a CH é reconhecida como a mais eficiente para atuar em contextos de crescimento exponencial dos dados, sendo capaz de diferenciar e rotular um volume muito grande de categorias e subcategorias. Não obstante, também é capaz de identificar instâncias ainda não rotuladas (ou desconhecidas) o que é um grande diferencial, dada a necessidade de adaptabilidade rápida com o dinamismo das mudanças (NAIK; RANGWALA, 2018).

O aprendizado através de CH possui diversos desafios que devem ser considerados para a construção de um classificador de chamados. O primeiro é que, ao contrário da classificação binária, em que cada documento (instância) pertence exclusivamente a apenas uma (única) classe, a hierárquica pode ser multirrótulo, ou seja, cada instância pode pertencer a várias classes de ramos completamente diferentes na hierarquia. O segundo desafio é a existência de muitas classes com poucas amostras. Esse problema pode afetar o aprendizado do modelo e reduzir a capacidade de generalização, tornando-o propenso ao sobreajuste. O terceiro desafio é conseguir incorporar relacionamentos hierárquicos durante o treinamento dos modelos para otimizar seu desempenho, uma vez que encontrar a melhor solução é uma tarefa difícil e não trivial. Outro obstáculo é a inconsistência na estrutura hierárquica, em outras palavras, a presença de relacionamentos inconsistentes entre categorias pais e filhos/irmãos na hierarquia (NAIK; RANGWALA, 2018).

Frente a este contexto, este trabalho apresenta um método que possa superar esses principais obstáculos relacionados à CH de textos, aplicado especificamente aos chamados de suporte de uma organização.

1.1 Hipótese e objetivos

Diante deste contexto, o objetivo deste trabalho foi desenvolver um classificador de chamados com capacidade de distinguir entre um grande número de categorias e subcategorias existentes e cujos temas destas categorias estão interligados, ou seja, não são completamente independentes entre si.

A hipótese assumida neste projeto é que é possível categorizar automaticamente chamados digitais escritos em linguagem natural. Para isso, foram empregadas técnicas de Processamento de Linguagem Natural (PLN) e métodos consolidados de categorização hierárquica de textos. Além disso, também foi analisada a possibilidade de automatizar o

processo de classificação dos dados não estruturados de maneira escalável, possibilitando a redução do tempo e frequência de manutenção do modelo. Em resumo, as principais contribuições deste trabalho são:

- Oferecer um estudo sobre as metodologias e desafios existentes relacionados a CH de textos;
- Apresentar uma análise do desempenho de modelos de CH utilizando uma base de dados textual rotulada.

1.2 Organização do texto

O presente trabalho está organizado da seguinte forma:

- O Capítulo 2 aborda a fundamentação teórica, contemplando os principais conceitos relacionados à classificação textual e suas variações metodológicas. São apresentadas as diferentes modalidades de classificação (binária, multiclasse, multirrótulo e hierárquica), com ênfase nas características específicas da Classificação Hierárquica (CH). O capítulo inclui também uma revisão sistemática da literatura, analisando as principais abordagens existentes no campo;
- O Capítulo 3 detalha a metodologia proposta, descrevendo o processo de pré-processamento textual e a abordagem hierárquica selecionada para o treinamento do modelo;
- O Capítulo 4 apresenta a avaliação experimental, incluindo a caracterização do conjunto de dados, a metodologia empregada, as configurações experimentais, as métricas de avaliação utilizadas e a análise dos resultados obtidos;
- O Capítulo 5 sintetiza as contribuições da pesquisa e indica direcionamentos para investigações futuras.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, são introduzidos os principais fundamentos associados à categorização de textos, em especial, os tópicos que envolvem pré-processamento de texto e métodos de AM para realizar a tarefa de categorização. Na Seção 2.1, são apresentados os princípios básicos e essenciais sobre o pré-processamento e preparação do texto. Métodos de representação computacional de textos são apresentados na Seção 2.2. Posteriormente, são introduzidos os tipos de classificação na Seção 2.3, bem como os principais métodos de classificação na Seção 2.4. Na Seção 2.5, são apresentadas as definições da CH e, na Seção seguinte 2.6 os principais fundamentos sobre estrutura e profundidade hierárquica. Por fim, na Seção 2.7 são apresentados os diferentes tipos de métodos e trabalhos relacionados sobre o tema.

2.1 Pré-processamento e preparação do texto

Com o alto volume de dados não estruturados disponível, a Categorização de Textos (CT), que envolve designar categorias previamente estabelecidas a documentos textuais, é um subcampo importante e aplicado em muitos contextos, desde a indexação de documentos até a geração automatizada de metadados e a organização de documentos. As categorias são comumente baseadas em assuntos ou tópicos, embora também possam ser determinadas por estilo (como gêneros) ou relevância (ZHAN; YOSHIDA; TANG, 2011).

Até o final dos anos 80, a abordagem mais popular para a CT era a engenharia do conhecimento, que envolvia a definição manual de regras para classificar documentos. No entanto, nos anos 90, a abordagem de AM ganhou popularidade. Esta abordagem permite a construção automática de um classificador de texto, capaz de aprender as características das categorias de interesse a partir de um conjunto de documentos pré-classificados. Tal abordagem oferece precisão comparável à dos especialistas humanos e economiza mão-de-obra especializada, pois não requer a intervenção de engenheiros do conhecimento ou especialistas de domínio para a construção do classificador (SEBASTIANI, 2001).

Para a implementação de quase todas as aplicações de Processamento de Linguagem Natural, é imprescindível executar fases que convencionalmente são denominadas de pré-processamento. Nestas fases, os textos selecionados passam por uma sequência de passos com o objetivo de extrair as informações mais relevantes e uniformizar os dados (CASELI; NUNES, 2023). Algumas tarefas comuns no pré-processamento incluem:

Setenciação (ou sentenciamento): procedimento de divisão do texto em sentenças, isto é, o processo de identificar as unidades textuais de processamento que determinam os limites de cada sentença (HAPKE; HOWARD; LANE, 2019). Este procedimento é

intrinsecamente complexo, uma vez que a ambiguidade inerente às línguas impede a total certeza de onde uma sentença se encerra (READ *et al.*, 2012).

Tokenização: divisão em unidades linguísticas elementares. Semelhante à setenciação, essa divisão é realizada com base na separação das palavras por meio de delimitadores. Nesse contexto, é essencial identificar os limites das palavras utilizando caracteres delimitadores, como espaços em branco ou símbolos de pontuação, incluindo ",", ":", ";", "-" e "." (FRAKES; BAEZA-YATES, 1992).

Normalização: tarefa de converter as palavras para alguma forma padrão, como por exemplo:

- Normalização lexical: substituir palavras/expressões abreviadas por suas formas canônicas (ALMEIDA *et al.*, 2016) como, por exemplo, "vc" para "você". Vale ressaltar que é necessário ter alguns cuidados para realizar essa operação para que a substituição não seja feita com abreviações com significados diferentes como, por exemplo, "rs" como abreviação de "risos" ser substituído como "Os youtubers do RS tem muito sotaque" para "Os youtube**risos** do RS tem muito sotaque".

- Conversão para caracteres minúsculos: para fins de uniformização, todo o texto costuma ser convertido para letras minúsculas. Este processo permite que palavras que se diferem como, por exemplo, "hepatite" e "Hepatite", sejam tratadas de forma equivalente. Contudo, mesmo nessa situação, é preciso ter cautela com nomes próprios e outras representações que possuem semântica associada ao uso de letras maiúsculas e minúsculas como, por exemplo, erroneamente descaracterizar a denominação do estado do Rio Grande do Sul ("RS") para a abreviação "rs" (UYSAL; GUNAL, 2014).

- Lematização: reduz as palavras a sua forma mais básica. Contudo, para realizar essa tarefa é necessário consultar os recursos que definem a estrutura e o significado das palavras, como um dicionário da língua. O desafio é entender o sentido correto das palavras que podem ter significados diferentes dependendo do contexto (MANNING; RAGHAVAN; SCHÜTZE, 2009). Por exemplo, na frase "Quem casa, quer casa." a palavra "casa" na primeira vez que aparece se refere ao verbo "casar", enquanto na segunda vez se refere ao substantivo "casa".

- Radicalização: uniformiza e diminui o vocabulário, convertendo as palavras para seus radicais e em alguns casos, pode levar a perda de informação (LOVINS, 1968). Exemplo: as palavras "certo", "certidão", "incerto", "certamente", "certificação", "certeiro" e "incerteza" possuem o mesmo radical "cert" e, portanto, tornam "cert" um bom candidato a subpalavra.

Remoção de caracteres de pontuação: a existência de pontuações como pontos de exclamação, apóstrofo, vírgula, gera ruídos nos textos, uma vez que possibilita criar distinção para palavras com mesma grafia (por exemplo, “oi” e “oi,”) (TABASSUM; PATIL, 2020). A definição de caracteres de pontuação pode variar conforme o método, linguagem de programação ou software utilizados. Neste contexto, como pontuação, o ponto final, a vírgula, a sequência de dois hífens (–) e as aspas (” ou “).

Remoção de *stopwords*: baseando-se na hipótese da distribuição, a remoção das *stopwords* possibilita deixar apenas as palavras relevantes, que representam melhor o conteúdo, e remover aquelas que mais se repetem. A partir da frase “o médico pessoal do argentino Diego”, por exemplo, após a remoção de *stopwords* fica “médico pessoal argentino Diego”, tendo sido removidos os *tokens* “o” e “do” (UYSAL; GUNAL, 2014).

Simplificação lexical: torna um texto mais fácil de compreender substituindo palavras difíceis ou raras por palavras mais simples e comuns (CASELI; NUNES, 2023).

Remoção de palavras raras: mesmo após a substituição de palavras raras ou complexas, se a frequência dessas palavras nos dados de treinamento for baixa, a probabilidade de contribuírem para a identificação da categoria do documento é reduzida. Portanto, tais palavras podem ser descartadas (WEISS; INDURKHYA; ZHANG, 2010).

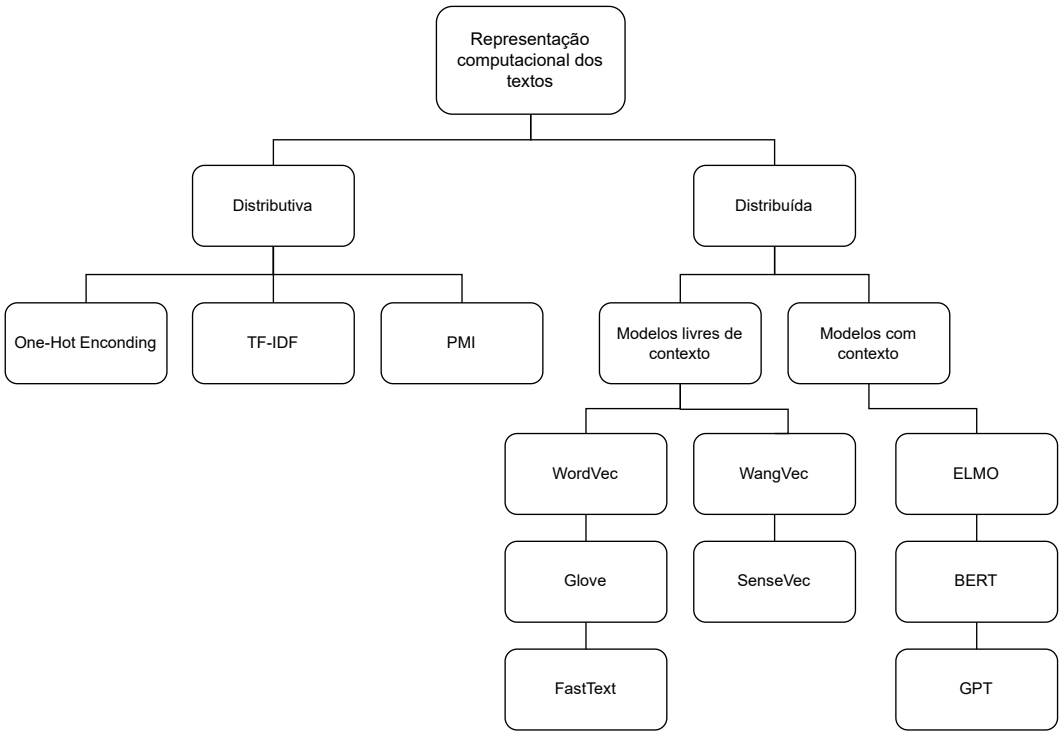
2.2 Representação computacional de textos

O Processamento de Linguagem Natural possibilita o uso de computadores para compreensão da comunicação humana, sendo uma ramificação da Inteligência Artificial (IA) (SRI, 2020). Como os algoritmos computacionais não possuem capacidade de processar símbolos ou palavras, foram desenvolvidas diversas técnicas que possibilitam representar numericamente um texto para realizar suas operações. Desta forma, para aplicar algoritmos de AM como, por exemplo, CH, é necessário transformar os textos em representações mais adequadas.

De acordo com Turian, Ratinov and Bengio (2010), as metodologias de transformação textual podem ser categorizadas em três abordagens principais. No escopo do presente trabalho, optou-se por concentrar a análise em duas destas categorias: a representação **distributiva** e a representação **distribuída**, cujas características e relações encontram-se ilustradas na Figura 2. Esta delimitação metodológica foi estabelecida considerando a adequação destas abordagens aos objetivos específicos desta pesquisa e suas respectivas capacidades de capturar as nuances semânticas necessárias para a tarefa de classificação hierárquica.

A **representação distributiva**, também conhecida como modelo espaço-vetorial, transforma cada palavra no documento em uma frequência de sua ocorrência dentro do texto. Desta forma, o vetor de representação indicará quando as palavras do vocabulário

Figura 2 – Tipos de métodos para representação computacional de textos



Fonte: Caseli and Nunes (2023)

estão presentes na mensagem. Um exemplo de representação distributiva é o *Bag-of-Words* (BoW) (CHOMSKY, 1957/2002).

Formalmente, seja $X = x_1, x_2, ..., x_n$ um conjunto com n documentos de textos, $T = t_1, t_2, ..., t_d$ o conjunto com d termos (atributos) que compõem um vocabulário predefinido e $Y = y_1, y_2, ..., y_k$ um conjunto finito de categorias ou rótulos do problema, o conjunto de documentos rotulados pode ser expresso como $D = (x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$, em que $x \in X$ e $y \subseteq Y$. Nos modelos de representação distributiva, cada documento pode ser representado por uma matriz que contém, em cada posição, um peso $w(t_i, x_j)$ que representa a frequência da ocorrência das palavras que compõem o documento (t_i em x). Esse peso pode variar de acordo com a metodologia de atribuição de pesos, sendo as mais comuns:

- **Binária (*One-Hot Encoding*)**: na representação binária, um termo/palavra é atribuído ao valor 1 (um) se estiver presente no documento e 0 (zero) se não estiver. É importante ressaltar que, nesta forma de representação, a frequência com que o termo aparece no documento não é levada em consideração. Em termos matemáticos, o peso binário pode ser calculado conforme ilustrado na Equação 2.1:

$$w(t_i, x) = \begin{cases} 1, & \text{se } t_i \text{ aparece em } x \\ 0, & \text{se } t_i \text{ não aparece em } x \end{cases} \tag{2.1}$$

- ***Term Frequency* (TF)**: o termo/palavra é quantificado com base em sua frequência de ocorrência de um termo t) em um documento d), estabelecendo uma relação diretamente proporcional entre o número de aparições do termo e seu peso na representação textual conforme ilustrado na Equação 2.2. Entretanto, a utilização isolada da métrica TF apresenta limitações na identificação dos termos mais relevantes de um documento, uma vez que determinadas palavras podem apresentar alta frequência em diversos documentos, reduzindo assim seu poder discriminativo na caracterização do conteúdo textual.

$$tf(t, d) = \frac{numero\_ocorrencias(t, d)}{total\_termos(d)} \tag{2.2}$$

- ***Inverse Document Frequency* (IDF)**: para mitigar o problema da métrica anterior, foi desenvolvida a métrica IDF, na qual é quantificada com base em sua frequência inversa de ocorrência de um termo t) em uma coleção de documentos N), conforme ilustrado na Equação 2.3. Em outras palavras, IDF mensura a raridade de uma palavra em um conjunto de documentos.

$$idf(t) = \log(\frac{N}{total\_documentos(t)}) \tag{2.3}$$

- ***Term Frequency-Inverse Document Frequency* (TF-IDF)**: o peso do termo/palavra é determinado multiplicando o TF (definido pela Equação 2.2) pelo IDF (definido pela Equação 2.3), conforme ilustrado na Equação 2.4. Isso significa que se uma palavra for comum em um documento específico, mas rara no contexto geral, pode indicar que a palavra é particularmente relevante para o conteúdo do documento em questão (WILBUR; KIM, 2009; RENNIE *et al.*, 2003).

$$tf\text{-}idf(t, d) = tf(t, d) \times idf(t) \tag{2.4}$$

- ***Pointwise Mutual Information* (PMI)**: medida estatística que permite mensurar a probabilidade de dois termos aparecem juntos em comparação com a probabilidade de aparecem separadamente. PMI foi introduzido por Church and Hanks (1990). Formalmente, o PMI entre um termo alvo x e um termo contexto y é definido pela Equação 2.5.

$$pmi(x,y) = \log_2 \frac{p(x,y)}{p(x)p(y)} = \log_2 \frac{p(x|y)}{p(x)} = \log_2 \frac{p(Y|x)}{p(y)} \tag{2.5}$$

onde a probabilidade de um termo $p(t)$ é calculada através da sua frequência divida pelo total de termos existentes em um documento, conforme ilutrado na Equação 2.6:

$$p(t) = \frac{frequencia(t)}{total\_termos} \tag{2.6}$$

No caso da BoW, como muitas palavras não aparecem em alguns documentos e existem muitas palavras sinônimas, esse modelo geralmente resulta em vetores grandes e esparsos, ou seja, com muitos zeros. Podendo assim, demandar alto custo computacional (XU *et al.*, 2013). Além disso, as representações distributivas têm problemas conhecidos, como perda de localidade de palavras, falta de informações semânticas e características sintáticas em diferentes contextos. Por exemplo, as frases "João é mais velho do que José" e "José é mais velho do que João" teriam representações idênticas, apesar de seus significados serem opostos. Assim, outras técnicas de representação computacional de textos surgiram para resolver estas questões, tais como técnicas de representação distribuída usando redes neurais (BENGIO *et al.*, 2006; COLLOBERT; WESTON, 2008; MNIH; HINTON, 2008; TURIAN; RATINOV; BENGIO, 2010; MIKOLOV *et al.*, 2013).

A **representação distribuída**, também chamada de representação vetorial ou *embedding*, é um método em que os termos/palavras são representadas por vetores semânticos. Esses vetores mostram, em cada dimensão, o significado das palavras com base em como elas são usadas nos textos (relações sintáticas e semânticas) (BENGIO *et al.*, 2006). Em contraste com a representação distributiva, os modelos de representação distribuída transformam os textos em números reais positivos ou negativos que representam relações semânticas e contextuais entre as palavras, em vez de contagens quase zero.

Por se basearem em distribuição, podem ser aprendidos automaticamente a partir de textos, sem a necessidade de supervisão humana. Os modelos aprendem através de algoritmos de AM, seja supervisionado ou não, por exemplo, usando redes neurais artificiais, ou, ainda, através da representação estatística da matriz de coocorrência de termos. Os modelos podem ser classificados como **livres de contexto** ou **com contexto** (WANG; HOU; CHE WANXIANGAND LIU, 2020):

- Modelos livres de contexto, como *Word2Vec* (MIKOLOV *et al.*, 2013), *GloVe* (PENNINGTON; SOCHER; MANNING, 2014), *FastText* (BOJANOWSKI *et al.*, 2017), *Wang2Vec* (LING *et al.*, 2015) e *SenseVec* (TRASK; MICHALAK; LIU, 2015), pois o vetor de uma mesma palavra é o mesmo independente do contexto que ela está inserida;

- Modelos com contexto, como *Embeddings from Language Models* (ELMo) (PETERS *et al.*, 2018), *Bidirectional Encoder Representations from Transformers* (BERT) (DEVLIN *et al.*, 2019) e *Generative Pre-trained Transformer* (GPT) (BROWN *et al.*, 2020), pois o vetor resultante de uma palavra depende do contexto que ela está inserida. Por apresentar a capacidade de interpretar as nuances semânticas, esta abordagem vem sendo muito utilizada para resolver problemas de classificação de textos.

A escolha entre essas abordagens depende das necessidades da aplicação (problema) e do método de classificação que será empregado, pois pode variar de acordo com a abordagem escolhida.

2.3 Classificação binária, multiclasse, multirrótulo e hierárquica

A ciência (e a arte) de ensinar computadores a aprender com dados, permitindo a criação de algoritmos, é conhecida como AM (GÉRON, 2019). De acordo com Samuel (1959), AM é o campo de estudo que possibilita aos computadores a habilidade de aprender sem explicitamente programá-los.

No campo do AM, os dispositivos computacionais são instruídos para adquirir conhecimento a partir de experiências anteriores (dados). Eles frequentemente utilizam um método de inferência conhecido como indução, que possibilita a obtenção de conclusões genéricas a partir de um conjunto específico de exemplos (FACELI *et al.*, 2011). Os dados de experiências anteriores (exemplos) E utilizados para adquirir aprendizado se chamam **conjuntos de treinamento** e cada exemplo de treinamento se chama **instância de treinamento (ou amostra)**. O computador aprende algum tipo de tarefa T e avalia o desempenho P para mensurar o quanto conseguiu aprender com os exemplos E (MITCHELL, 1997). AM é particularmente útil para:

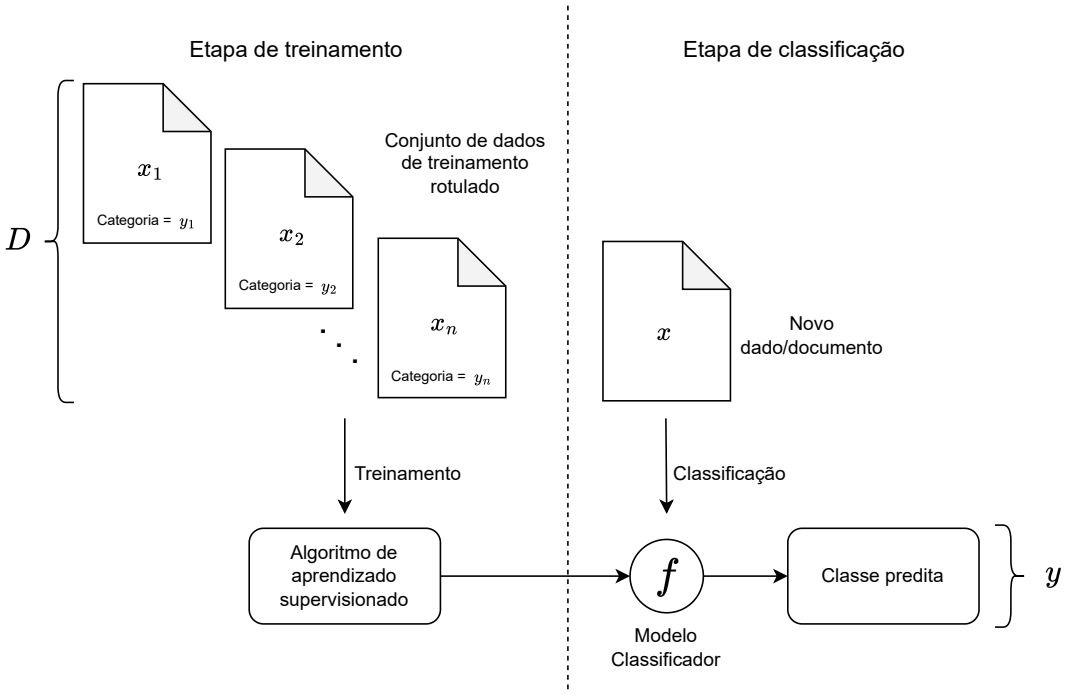
- Problemas cujas soluções atuais demandam uma quantidade significativa de ajustes ou extensas listas de regras (complexidade de manutenção): um algoritmo de AM pode simplificar o processo e geralmente apresenta um desempenho superior ao de métodos tradicionais.

- Problemas complexos que não possuem uma solução eficaz através de abordagens convencionais: as técnicas avançadas de AM podem ser capazes de encontrar uma solução. Por exemplo, a tarefa de reconhecimento de voz é altamente complexa, pois existem milhares de palavras em diversas línguas que variam de pronúncia de acordo com diferentes pessoas. Em vez de criar regras, a solução mais viável, até o momento, é criar algoritmos que aprendam sozinhos através de gravações.

- Adaptação a ambientes variáveis: um sistema baseado em AM tem a capacidade de se adaptar a novos dados através de treinamentos contínuos.
- Compreensão de problemas complexos e manipulação de grandes volumes de dados.

A tarefa de classificação de texto começa com um conjunto de treinamento $D = (x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$ de documentos rotulados, em que $x \in X$ e $y \subseteq Y$, com categorias $Y = y_1, y_2, ..., y_k$ (por exemplo, esportes, política). O objetivo é treinar um modelo de classificação f (classificador) conforme ilustrado na Figura 3, capaz de atribuir automaticamente rótulo(s) de classe correto(s) a um novo documento x cujas classes são desconhecidas (ZHAN; YOSHIDA; TANG, 2011).

Figura 3 – Lógica de criação de algoritmos de aprendizado supervisionado



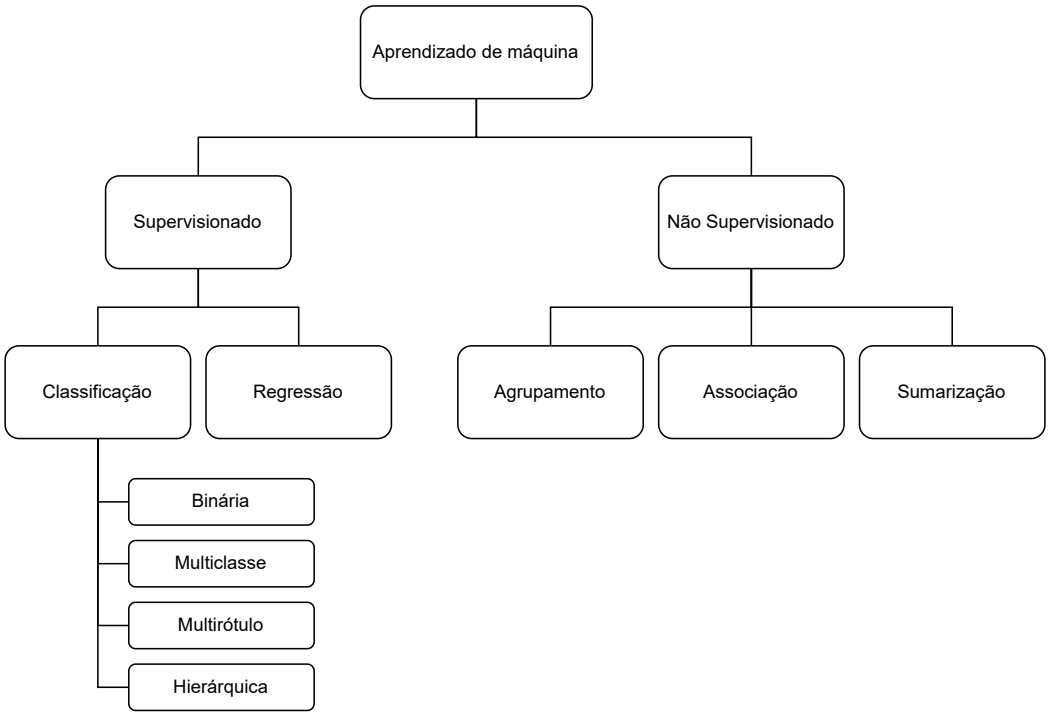
Fonte: Bittencourt (2020)

A classificação faz parte de uma categoria dentro de AM, pois os dados de treinamento que são fornecidos ao algoritmo incluem as soluções desejadas, chamadas de rótulos/categorias (GÉRON, 2019). Tarefas de classificação podem ser divididas em: binárias, multiclasse, multirrótulo e hierárquicas (conforme ilustrado na Figura 4).

Classificadores binários fazem distinções entre apenas duas classes (GÉRON, 2019). Por exemplo, diagnosticar um paciente se ele possui ou não uma doença com base nos sintomas que apresenta.

Classificadores multiclasse, também conhecidos como classificadores multinomiais, fazem distinções entre mais de duas classes. Um exemplo prático é a classificação de notícias em diversas categorias como: esporte, política, saúde, etc.

Figura 4 – Hierarquia de Aprendizado de Máquina



Fonte: Faceli *et al.* (2011)

Classificadores multirrótulo, podem atribuir mais de uma classe para a mesma instância. Por exemplo, um modelo de identificação de comentários ofensivos em fóruns de discussão, uma amostra poderia ser atribuída à classe machismo e racismo ao mesmo tempo.

Classificadores hierárquicos, as classes podem se organizar em uma relação de taxonomia ou dependência. Desta forma, uma instância pode ser atribuída a uma subclasse dentro de uma superclasse. Um exemplo prático é a classificação de artigos de notícias, sendo uma amostra atribuída à superclasse esportes e à subclasse futebol.

Na literatura, existem diversas metodologias para resolver problemas de CH que podem ser divididas em três categorias: plana, local e global. A classificação plana é a mais simples, pois desconsidera a estrutura hierárquica para o treinamento do modelo. Já a classificação local divide a estrutura hierárquica em várias estruturas menores e utiliza as relações locais. Por fim, a global considera toda a hierarquia de classes para treinamento (NAIK; RANGWALA, 2018).

2.4 Aprendizado supervisionado - Classificação

Existem vários métodos de classificação disponíveis na literatura e devem ser escolhidos de acordo com a aplicação/tarefa a ser realizada. Os mais tradicionais incluem (FACELI *et al.*, 2011):

- Métodos baseados em distâncias: a hipótese é que dados similares tendem a estar próximos em uma região e distantes de dados divergentes. Um exemplo de método de distância é *K-Nearest Neighbours* (KNN) (COVER; HART, 2006);

- Métodos probabilísticos: a hipótese é que a probabilidade de um evento A (por exemplo, um paciente ter uma determinada doença) dado um evento B (por exemplo, o paciente ter um resultado positivo em um exame) não depende apenas da relação entre A e B (MITCHELL, 1997). Isso significa que, ao calcular a probabilidade, esses métodos levam em conta tanto a informação específica do exame quanto a informação geral sobre a frequência da doença. Isso pode ajudar a fazer previsões mais precisas em situações complexas onde vários fatores estão em jogo. Um exemplo de método probabilístico é o *Naive Bayes* (NB) (MCCALLUM; NIGAM, 1998);

- Métodos simbólicos: são métodos que buscam representar os dados de maneira simbólica, facilitando a interpretação e compreensão do processo de classificação para seres humanos. É o caso das *Decision Trees* (DT) e *Classification and Regression Trees* (CART) (BREIMAN *et al.*, 1984);

- Métodos conexionistas: método basea-se na busca de aproximar a maneira como o processo do cérebro humano opera para classificar os dados. O cérebro é composto por células denominadas neurônios, que intercambiam informações através de conexões chamadas sinapses. Cada neurônio, estando conectado a vários outros, possibilita a atuação de muitos em paralelo, o que é crucial para o funcionamento do cérebro. Desta forma, o funcionamento do cérebro inspirou no desenvolvimento dos métodos conexionistas de AM, as Redes Neurais Artificiais (RNAs) (HAYKIN, 1998);

- Métodos baseados em maximização de margens: método que busca, através de diversas técnicas, separar dados que pertencem a classes distintas maximizando-se as margens, determinando o grau em que os dados estão separados (SMOLA *et al.*, 2000). Um exemplo de método é *Support Vector Machines* (SVM) (CORTES; VAPNIK, 1995);

- Métodos baseados em modelos múltiplos preditivos: método que utiliza um conjunto de preditores, cada qual as decisões individuais dos modelos são combinadas ou agregadas para classificar os dados (DIETTERICH, 1997). Exemplos de modelos: *Random Forest* (RF) (BREIMAN, 2001) e o *boosting* adaptativo (FREUND; SCHAPIRE, 1996).

Além dos métodos citados acima, existem diversos outros tipos de classificadores e regressores. Por ser o foco deste estudo, a Classificação Hierárquica será detalhada na próxima seção.

2.5 Definição de Classificação Hierárquica

CH foi explorada por diversos pesquisadores como, Koller and Sahami (1997), Larkey (1998), McCallum *et al.* (1998), D'Alessio *et al.* (2000), Vaithyanathan, Mao and Dom (2000), Dumais and Chen (2000), Sun and Lim (2001), Cai and Hofmann (2004). Essa metodologia foi amplamente estudada e utilizada por apresentar maior eficiência e escalabilidade em casos de crescimento exponencial de dados, principalmente para fins de organização de documentos (CHAKRABARTI *et al.*, 1998).

Isso se deve pelo fato de que a CH possibilita acessar e gerenciar as categorias de forma que facilite na recuperação de documentos, uma vez que permite especializar ou generalizar determinado assunto ao se aproximar da raiz ou nós-folhas. Foi e ainda é amplamente utilizada como, por exemplo, pelo *Yahoo!* e *Wikipedia* para pesquisas de suas páginas web (ANICK; VAITHYANATHAN, 1997; PARTALAS *et al.*, 2015), pela *International Business Machines - IBM* para organizar suas patentes (APTÉ; DAMERAU; WEISS, 1994), pela *Columbia University Libraries* para organizar seu catálogo de livros online (DAVIS, 2002), para classificação das emoções na fala/discurso (XIAO *et al.*, 2007) e, mais recentemente, com classificação de tópicos e subtópicos para recuperação de documentos em chatbots (RODRÍGUEZ-CANTELAR *et al.*, 2023).

CH é um dentre diversos tipos de Classificação Estruturada (*Structured Classification* - SC) do qual as classes são organizadas de maneira hierárquica (SILLA; FREITAS, 2011). Vale ressaltar que SC possui outros tipos que não se enquadram nessa estrutura como, por exemplo, classes que possuem organização temporal ou sequencial (*Label Sequence Learning*) (ALTUN; HOFMANN, 2003; TSOCHANTARIDIS *et al.*, 2005). Antes de estabelecer a estrutura das classes, é fundamental definir os tipos de nós presentes na hierarquia.

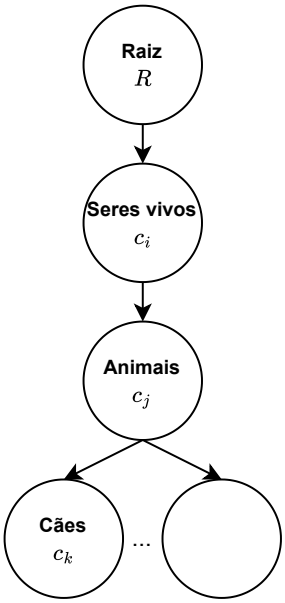
Definição 1: define-se como nó raiz na hierarquia o nó que não possui um nó ancestral (nó pai) e pelo menos um nó descendente (nó filho). Vale ressaltar que existem hierarquias complexas que possuem mais de um nó raiz.

Definição 2: define-se como nó interno, também é conhecido como nó não terminal, o nó que possui pelo menos um nó ancestral (nó pai) e pelo menos um nó descendente (nó filho). Ou seja, o nó interno é intermediário, conectando nós superiores e inferiores. Conforme ilustrado na Figura 5 o nó "Animais" é considerado como nó interno na hierarquia.

Definição 3: define-se como nó folha, também é conhecido como nó terminal, o nó que possui pelo menos um nó ancestral (nó pai) e nenhum nó descendente (nó filho). Em outras palavras, nó que não pertence à raiz ou ao nó interno é o nó folha. Conforme ilustrado na Figura 5 o nó "Cães" é considerado como nó interno na hierarquia.

A organização hierárquica das classes, também chamada de taxonomia de classes, foi

Figura 5 – Propriedade de transitividade



Fonte: Silla and Freitas (2011)

inicialmente definida por Wu, Zhang and Honavar (2005) da seguinte forma: a taxonomia de classes é uma **estrutura em árvore** sobre um conjunto parcialmente ordenado $(C, \prec)$, no qual C é um conjunto finito das categorias de um determinado domínio e $\prec$ representa a relação *"IS-A"* como anti-reflexiva e transitiva. Entretanto, a definição feita por Wu, Zhang and Honavar (2005) foi reexaminada por Silla and Freitas (2011) e definiu a relação *"IS-A"* como:

1. Um único grande elemento R é a raiz da árvore.
2. Assimétrica: $\forall c_i, c_j \in C$, se $c_i \prec c_j$ então $c_j \not\prec c_i$
3. Anti-reflexiva: $\forall c_i \in C$, $c_i \not\prec c_i$
4. Transitiva: $\forall c_i, c_j, c_k \in C$, $c_i \prec c_j$ e $c_j \prec c_k \Rightarrow c_i \prec c_k$

A propriedade 1 indica que a raiz da árvore R deve ser uma única categoria R que antecede as demais. Quando a instância é uma raiz, seu rótulo precisa ser geral e aplicável a qualquer outra instância mais específica abaixo dela (WU; ZHANG; HONAVAR, 2005).

A propriedade 2 denota que a relação entre as classes é assimétrica, uma vez que se a categoria c_i antecede a categoria c_j e, portanto, a categoria c_i possui uma relação *"IS-A"* com c_j. Contudo, a mesma relação inversa entre as categorias não deve ocorrer para que a organização da hierarquia seja válida. Ruiz and Srinivasan (2002) fornece o seguinte exemplo para melhor entendimento: todos os cães são animais, mas nem todos os animais são cães.

A propriedade 3 expressa o princípio da anti-reflexidade, na qual a categoria c_i não pode anteceder a si mesma. Tal propriedade, em conjunto com a anterior, possibilita que as classes não tenham uma relação cíclica entre si, possibilitando apenas estruturas hierárquicas (WU; ZHANG; HONAVAR, 2005; SILLA; FREITAS, 2011).

A propriedade 4 define o princípio da transitividade das relações de antecedência entre as classes. Ou seja, se a categoria c_i antecede a categoria c_j e a mesma categoria antecede c_k, então há uma relação de antecedência de c_i e c_j (SILLA; FREITAS, 2011). Utilizando o mesmo exemplo dado anteriormente, todos os animais são seres vivos e todos os cães são animais, logo, todos os cães são seres vivos (conforme ilustrado na Figura 5).

Desta forma, qualquer problema classificatório no qual as classes se adequem às quatro regras de relação *"IS-A"* citadas acima pode ser considerado um problema de CH (SILLA; FREITAS, 2011).

2.6 Tipos de estrutura e profundidade da Classificação Hierárquica

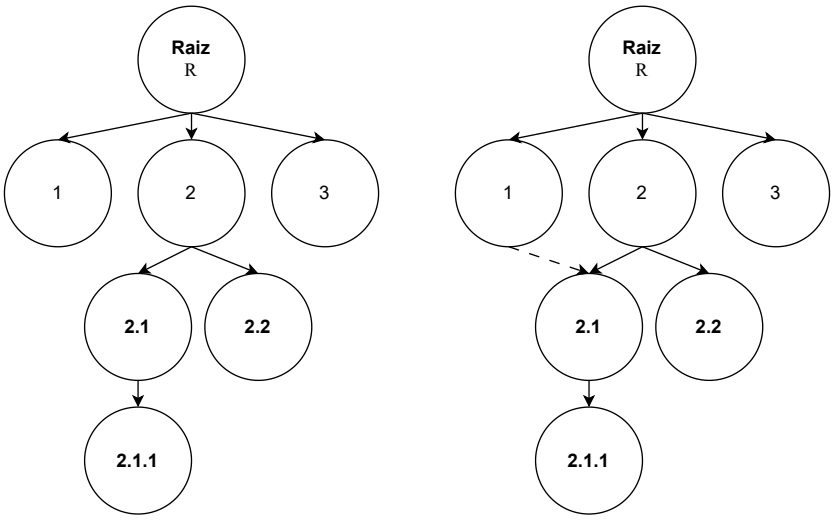
A estrutura hierárquica pode ser dividida em dois tipos: Árvore e Grafos Acíclicos Dirigidos (*Direct Acyclic Graphs - DAG*). A diferença entre as duas estruturas está no fato de que a DAG permite que as categorias tenham mais de um antecessor em suas relações (SILLA; FREITAS, 2011; FACELI *et al.*, 2011). Ou seja, um documento pode ser classificado em mais de uma categoria (ROUSU *et al.*, 2006). Em contrapartida, na estrutura em árvore, cada categoria deve possuir apenas um único antecessor, conforme ilustrado na Figura 6.

Existem dois caminhos possíveis para realizar a CH: classificação única ou multirrótulo (*multilabel*). Na classificação única, um documento pertence a uma categoria exclusiva, enquanto na classificação multirrótulo, um documento pode pertencer a várias categorias, das quais podem pertencer a múltiplas categorias antecedentes da hierarquia (TIKK; BIRÓ; YANG, 2003). Rousu *et al.* (2006) define a CH multirrótulo como "uma união de caminhos parciais na hierarquia".

A partir desta definição, qualquer problema de CH pode ser considerado multirrótulo. Para exemplificar, conforme ilustrado na Figura 6 se um documento pertencer à classe 2.1.1, pode-se afirmar que este mesmo documento pertence às classes 2.1 e 2, possuindo três classes para um único documento.

Outro aspecto relevante dos problemas de CH está relacionado com a profundidade das rotulações, podendo ser implementadas de duas maneiras: classificando apenas no último nível da hierarquia, conhecido em inglês como *Mandatory Leaf-Node Prediction* (MLNP) (FREITAS; CARVALHO, 2007) e também pode ser chamado de *Virtual Category Tree* (SUN; LIM, 2001); ou classificando em qualquer nível da hierarquia *Non-Mandatory Leaf Node Prediction* (NMLNP) (FREITAS; CARVALHO, 2007).

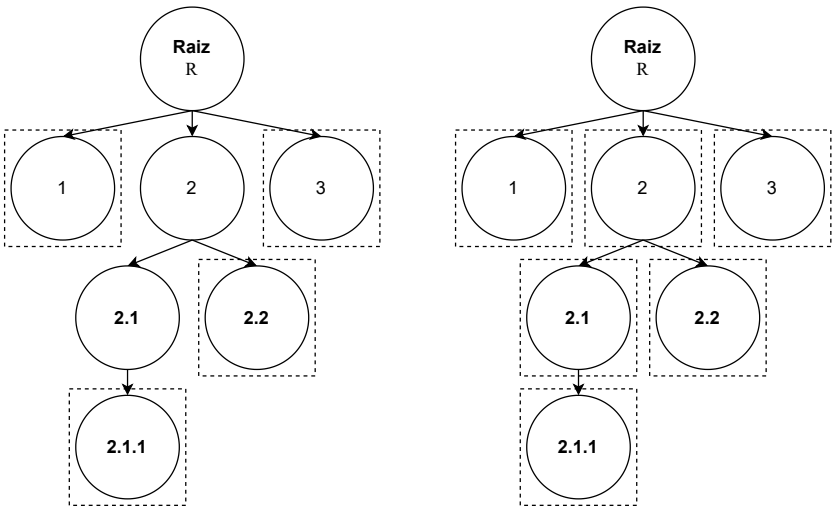
Figura 6 – Tipos de estruturas hierárquicas



Árvore DAG

Fonte: Silla and Freitas (2011)

Figura 7 – Profundidade das rotulações



MLNP NMLNP

Fonte: Naik and Rangwala (2018)

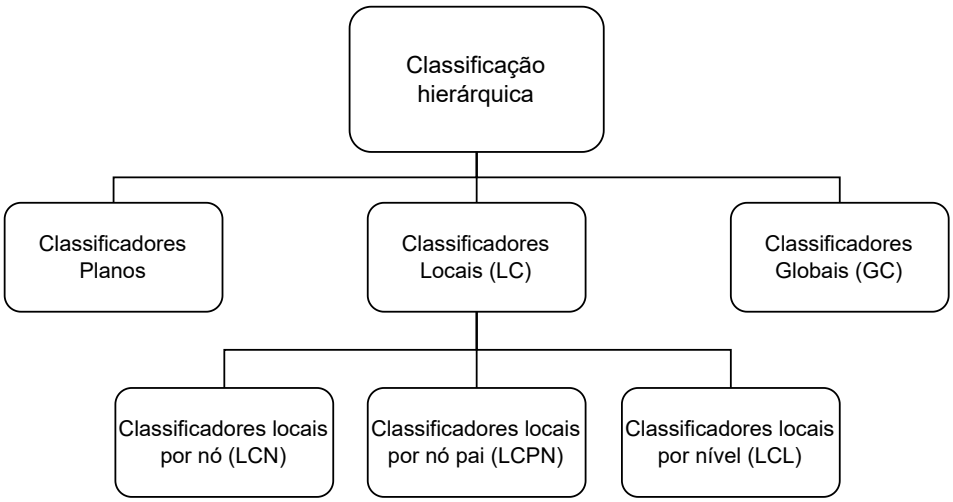
Conforme ilustrado na Figura 7, na CH por MLNP, seria possível obter apenas as categorias 1, 2.2, 2.1.1 e 3. Em contrapartida, utilizando o método NMLNP, os dados poderiam ser classificados em qualquer nível 1, 2, 2.1, 2.2, 2.1.1 e 3 sendo possível utilizar um limiar (*threshold*) para determinar a profundidade desejada.

2.7 Tipos de Classificação Hierárquica

Os métodos de CH são definidos pela forma como utilizam a estrutura hierárquica para construção do classificador e para rotular novos dados. Existem três tipos de abordagens desenvolvidas até o momento: classificação plana, local e global (conforme ilustrado na Figura 8). Nas Subseções seguintes, são apresentadas as definições e abordagens

desenvolvidas aplicadas a problemas de CH de textos.

Figura 8 – Tipos de classificação hieráquica



Fonte: Autor

2.7.1 Classificação Plana

A Classificação Plana (*Flat Classification*), segundo Silla and Freitas (2011) é a abordagem mais simples para lidar com problemas de CH, rotulando-se apenas as classes do último nível da hierarquia. Essa técnica é aplicável para ambos os tipos de estruturas hierárquicas (árvore ou DAG), pois ignora as classes antecedentes, seguindo a abordagem MLNP.

A principal vantagem desta técnica reside na redução de complexidade de implementação, permitindo o uso de algoritmos de classificação não hierárquicos. Além disso, possibilita atribuir as classes antecedentes a uma mesma instância, seguindo a relação das classes de uma hierarquia. Entretanto, esse método traz diversas desvantagens, uma vez que ao rotular os níveis mais baixos da hierarquia resulta em um aumento substancial das classes-alvo, potencialmente desequilibradas. O classificador plano precisa tomar uma decisão difícil e não trivial de escolher apenas uma única classe dentre todas, ignorando as relações entre os níveis da hierarquia (SILLA; FREITAS, 2011; BABBAR *et al.*, 2013).

2.7.2 Classificação Local

Classificação Local (*Local Classification*), segundo Faceli *et al.* (2011) é uma abordagem que utiliza informações locais da hierarquia para classificação, possibilitando rotular em qualquer nível da hierarquia (NMLNP), com exceção da raiz. Em virtude disso, esse método pode ser aplicado a problemas de classificação multirrótulo (DAG). Outra vantagem é a possibilidade de utilizar algoritmos de classificação não hierárquicos.

Contudo, esta técnica não é adequada quando há um grande volume de classes, pois se torna ineficiente para lidar com o desbalanceamento dos dados de treinamento. Outro problema está relacionado à propagação de erros na rotulação de classes antecedentes para os níveis mais baixos da hierarquia (IRSAN; KHODRA, 2016). Segundo Silla and Freitas (2011), existem três maneiras de classificar localmente: por nó, por nó pai ou por nível.

2.7.2.1 *Local Classifier per Node*

Local Classifier per Node (LCN), é uma técnica que combina algoritmos classificadores binários em cada nó da estrutura hierárquica, com exceção da raiz R. Esta abordagem tem como principal característica a versatilidade para determinar quais amostras serão consideradas positivas e negativas dos nós não-folha. Em estudos anteriores de EISNER *et al.* (2005), FAGNI; SEBASTIANI (2007) e CECI; MALERBA (2007), destacam-se seis regras (comumente chamadas de "políticas") para definir os exemplos positivos e negativos da amostra. A notação empregada nestas definições segue a convenção proposta por Fagni and Sebastiani (2007), detalhada na Tabela 2.

Tabela 2 – Notação para exemplos positivos e negativos de treinamento

Símbolo	**Significado**
Tr	Conjunto de todos dos exemplos de treinamento
$Tr^{+}(c_j)$	Conjunto de todos dos exemplos positivos de (c_j)
$Tr^{-}(c_j)$	Conjunto de todos dos exemplos negativos de (c_j)
$\uparrow (c_j)$	Categoria pai de (c_j)
$\downarrow (c_j)$	Conjunto de todas as categorias filho de (c_j)
$\Uparrow (c_j)$	Conjunto de todas as categorias antecessoras de (c_j)
$\Downarrow (c_j)$	Conjunto de todas as categorias descendentes de (c_j)
$\leftrightarrow (c_j)$	O conjunto de categorias de irmãos de (c_j)
$*(c_j)$	Denota exemplos cuja classe conhecida mais específica é (c_j)

- Política **"exclusiva"** (*"exclusive" policy*) proposta por Eisner *et al.* (2005), restringe-se aos exemplos explicitamente rotulados pelo nó (c_j) para definir a classe mais específica como positiva $Tr^{+}(c_j) = *(c_j)$, utilizando os demais nós como negativos $Tr^{-}(c_j) = Tr \setminus *(c_j)$. Conforme ilustrado na Figura 9, esta abordagem ignora a estrutura hierárquica e os nós descendentes $\Downarrow (c_j)$ são considerados como negativos, o que contradiz a definição da relação *"IS-A"* da hierarquia. A aplicabilidade deste método é limitada a problemas com hierarquias parcialmente conhecidas, onde apenas os níveis mais superficiais são rotulados.

- Política **"menos exclusiva"** (*"less exclusive" policy*) também proposta por Eisner *et al.* (2005), considera os exemplos rotulados pelo nó (c_j) como positivos $Tr^{+}(c_j) = *(c_j)$, enquanto os demais nós, com excessão dos descendentes do nó (c_j), como negativos $Tr^{-}(c_j) = Tr \setminus *(c_j) \cup \Downarrow (c_j)$. Ilustrado na Figura 10, esta

abordagem resolve o problema da metodologia anterior, no qual os nós descendentes são tomados como exemplos negativos. Entretando, a aplicabilidade se mantém restrita a rotulagem superficial da hierarquia.

- Política **"menos inclusiva"** (*"less inclusive" policy*), proposta por Eisner *et al.* (2005), também pode ser encontrada na literatura como política **"tudo"** (*"ALL" policy*) definida por Fagni and Sebastiani (2007). Conforme ilustrado na Figura 11, esta abordagem determina que os exemplos do nó (c_j) e todos os seus descendentes $\Downarrow (c_j)$ são rotulados como a classe positiva $Tr^+(c_j) = *(c_j) \cup \Downarrow (c_j)$. Os exemplos rotulados como negativos são os nós restantes da hierarquia e seus descendentes $Tr^-(c_j) = Tr \setminus * (c_j) \cup \Downarrow (c_j)$.

- Política **"inclusiva"** (*"inclusive" policy*), proposta por Eisner *et al.* (2005), determina que o nó (c_j) e todos os seus descendentes $\Downarrow (c_j)$ são rotulados como a classe positiva $Tr^+(c_j) = *(c_j) \cup \Downarrow (c_j)$. Todos os demais nós, com excessão dos ancestrais $\Uparrow (c_j)$, são rotulados como negativos $Tr^-(c_j) = Tr \setminus * (c_j) \cup \Downarrow (c_j) \cup \Uparrow (c_j)$, conforme ilustrado na Figura 12.

- Política de **"irmãos"** (*"siblings" policy*), inicialmente proposta por Wiener, Pedersen and Weigend (1995) e, posteriormente, definida por Fagni and Sebastiani (2007) e citada por Ceci and Malerba (2007) como "conjuntos de treinamento hierárquicos" (*"hierarchical training sets"*). Conforme ilustrado na Figura 13, esta abordagem considera o nó (c_j) e todos os seus descendentes $\Downarrow (c_j)$ como exemplos positivos $Tr^+(c_j) = *(c_j) \cup \Downarrow (c_j)$. Já os exemplos negativos é tomado pelo conjunto de categorias de irmãos de (c_j) e seus descendentes $Tr^-(c_j) = \leftrightarrow (c_j) \cup \Downarrow (\leftrightarrow (c_j))$.

- Política de **"irmãos exclusivos"** (*"exclusive siblings" policy*), definida por Ceci and Malerba (2007) como "conjuntos de treinamento apropriados" (*"proper training sets"*). Esta abordagem considera apenas o nó (c_j) como exemplos positivos $Tr^+(c_j) = *(c_j)$ e seus irmãos como negativos $Tr^-(c_j) = \leftrightarrow (c_j)$, ignorando os todos os demais nós, conforme ilustrado na Figura 14.

Independente da política para definir os exemplos positivos e negativos de cada nó, o processo de classificação é normalmente realizado *top-down*, começando pela raiz até chegar aos níveis mais baixos (KOLLER; SAHAMI, 1997; NAIK; RANGWALA, 2018). Entretanto, não é uma regra obrigatória a ser seguida. Na literatura, há trabalhos que afirmam que a abordagem LCN e a abordagem *top-down* são a mesma abordagem, mas são abordagens distintas que frequentemente são utilizadas em conjunto (SILLA; FREITAS, 2011).

A LCN apresenta uma característica multirrótulo, possibilitando a rotulação simultânea de diversas categorias em cada nível de classe. No entanto, como mencionado

Figura 9 – Exclusiva

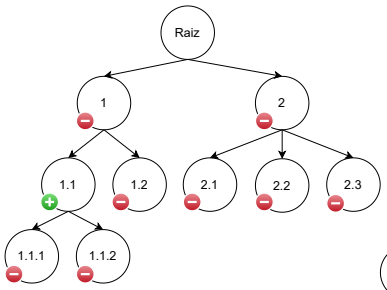


Figura 10 – Menos exclusiva

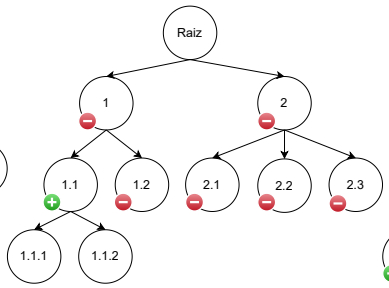


Figura 11 – Menos inclusiva

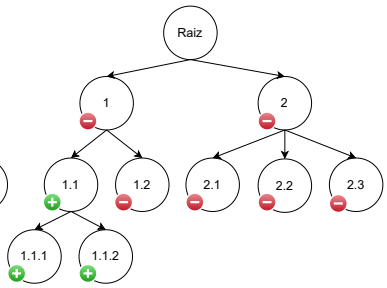


Figura 12 – Inclusiva

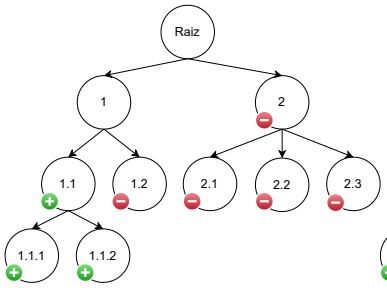


Figura 13 – Irmãos

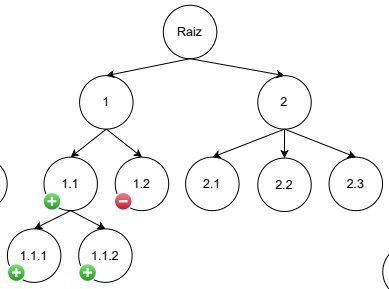
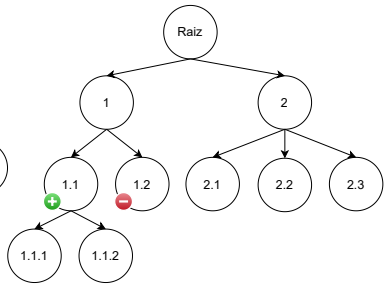


Figura 14 – Exclusiva de irmãos



Fonte: Metz (2011)

anteriormente, essa abordagem pode não preservar a relação hierárquica, dependendo da estrutura da hierarquia e da política utilizada, o que pode gerar inconsistências nas previsões das classes em diferentes níveis. Existem trabalhos que propõem formas para tratar tais inconsistências de maneira mais profunda como, por exemplo, Silla and Freitas (2011).

Diversos estudos na literatura empregam a abordagem LCN para fins de categorização de textos. Bennett and Nguyen (2009) realizaram o balanceamento das classes na fase de treinamento para redução de ruídos que o desbalanceamento causa. Além disso, utilizaram a abordagem *bottom-up*, na qual foram utilizadas as informações das saídas de cada classificador dos nós inferiores para classificação de cada um dos nós superiores como atributo. Por fim, é realizada a classificação final *top-down* para cada nó da hierarquia, assim mitigando potenciais inconsistências na hierarquia. A pesquisa foi aplicada a uma base com conteúdos textuais de web sites rotulados *Open Directory Project's* (ODP).

Ramírez-Corona, Sucar and Morales (2016) utilizaram uma abordagem diferente para lidar com a propagação de inconsistências, calculando a correlação entre o nó e seus ancestrais como atributo adicional. Em seguida, calcularam uma pontuação para cada caminho e, por fim, realizaram a rotulação final para cada nó da hierarquia. Foram utilizadas 20 bases de dados, entre elas bases de dados textuais de sequências genômicas (*FunCat*) e ontologia genética (*Gene Ontology*).

Cesa-Bianchi, Gentile and Zaniboni (2006a) utilizaram a medida de desempenho específica para problemas de CH (*H-loss*) e a abordagem *bottom-up* para reduzir os impactos da propagação de inconsistências. Posteriormente, Cesa-Bianchi, Gentile and Zaniboni (2006b) utilizaram a mesma medida de desempenho, combinada a um novo algoritmo que é treinado incrementalmente a cada nó *top-down*. Obteve-se desempenho próximo a *Support Vector Machines* (SVM) hierárquicas e aplicável a problemas com caminhos múltiplos e/ou parciais na hierarquia.

Banerjee *et al.* (2019) desenvolveram uma nova abordagem, chamada de "transferência de aprendizado" (*"transfer learning"*), uma vez que reutiliza os parâmetros do algoritmo treinado para o nó pai e ajustados para classificar os nós filhos. A primeira vantagem apresentada nos resultados deste estudo é a melhora na classificação de categorias que possuem poucos exemplos e a segunda vantagem é obter melhores resultados utilizando classificadores binários em vez de classificadores multirrótulo.

Silva, Suguiy and Silla (2023) exploraram o problema de classificação de gênero musical utilizando LCN, que até então foi pouco investigado em uma estrutura hierárquica de classificação. Foram comparados quatro tipos de políticas ("exclusiva", "menos exclusiva", "inclusiva" e "menos inclusiva") sobre uma base de músicas disponibilizadas e rotuladas pelos próprios artistas (*Free Music Archive Database*). A política "menos inclusiva" apresentou o melhor resultado (de Medida-F), comparando-a com as demais políticas.

Miranda, Köhnecke and Renard (2023) compararam classificadores locais com classificadores planos, sobre uma base textual de reclamações de consumidores (*Consumer Financial Protection Bureau of the United States*) categorizando em produto e sub-produto utilizando a biblioteca HiClass. Os resultados dos LCN apresentaram maior performance (de Medida-F), comparando-a com classificadores planos.

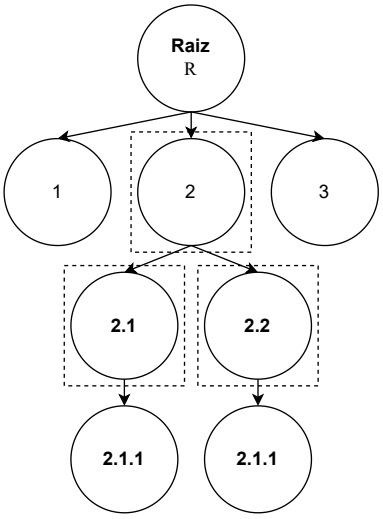
2.7.2.2 *Local Classifier per Parent Node*

Local Classifier per Parent Node (LCPN) é uma técnica na qual o algoritmo é treinado para classificar os rótulos dos nós pais na hierarquia, conforme ilustrado na Figura 15. Esta abordagem também pode ser aplicada a problemas multirrótulo e, portanto, pode possuir as mesmas inconsistências de LCN (SILLA; FREITAS, 2011).

Nesta abordagem, existem duas políticas para determinar quais exemplos serão definidos como positivos e negativos: Política de **"irmãos"** e Política de **"irmãos exclusivos"**. A diferença entre as técnicas está no uso dos nós descendentes ou apenas nós filhos imediatos como exemplos para classificação (PEREIRA; COSTA; JR., 2021).

Koller and Sahami (1997) foram os primeiros a propor o primeiro método de LCPN, adotando a estratégia *top-down* na fase de teste. Esta abordagem caracteriza-se pela

Figura 15 – *Local Classifier per Parent Node* (LCPN)



Fonte: Silla and Freitas (2011)

aplicação de um mesmo algoritmo de classificação em todos os níveis da hierarquia de classes (SILLA; FREITAS, 2011).

Posteriormente, Secker *et al.* (2007) desenvolveram uma abordagem diferente utilizando o método chamado "classificador seletivo" (*"selective classifier"*), no qual são aplicados diferentes algoritmos para cada nó pai da hierarquia. Para determinar qual classificador deve ser utilizado em cada nó pai, conjunto de treinamento é subdividido de maneira aleatória em dois subconjuntos menores: treinamento e validação. O classificador escolhido para cada nó pai é o que possui maior precisão no conjunto de validação. Além disso, essa abordagem também utiliza a estratégia *top-down* na etapa de teste.

Em seguida, uma nova proposta de abordagem foi desenvolvida por Holden and Freitas (2008) para o "classificador seletivo". Diferentes algoritmos continuam sendo aplicados a cada nó da hierarquia, mas a avaliação é realizada de forma global. Desta forma, a combinação de classificadores a cada nó pai considera toda a taxonomia de uma só vez na avaliação do desempenho.

Krendzelak and Jakab (2019) aplicou LCPN utilizando apenas um algoritmo por nó pai. Cada algoritmo foi treinado de forma independente utilizando apenas como exemplos os nós pai e seus descendentes. O estudo apresentou bons resultados utilizando *Convolutional Neural Networks* (CNNs) em conjunto com LCPN, superando outras abordagens locais.

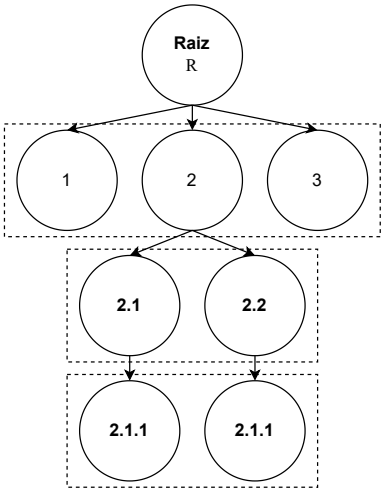
2.7.2.3 *Local Classifier per Level*

Local Classifier per Level (LCL), é uma técnica na qual um classificador plano é criado para cada nível da hierarquia, conforme ilustrado na Figura 16. Semelhante à abordagem anterior (LCPN), pode-se utilizar todos os nós descendentes ou apenas os nós-filho diretos para definir as amostras positivas e negativas no treinamento do

classificador.

Esta técnica possui a vantagem de inibir inconsistências horizontais, visto que só é possível rotular as classes do mesmo nível. Entretanto, é possível ter inconsistências verticais, visto que os classificadores realizam previsões independentes por nível. Outra desvantagem é o aumento substancial das classes quando a hierarquia possui muitas ramificações profundas. Freitas and Carvalho (2007) mencionou esta técnica como uma possibilidade para lidar com problemas de CH. Contudo, é pouco utilizada na literatura, comparando-a com as demais abordagens (SILLA; FREITAS, 2011).

Figura 16 – *Local Classifier per Level* (LCL)



Fonte: Silla and Freitas (2011)

Com o objetivo de tornar esta abordagem útil, os trabalhos de Clare and King (2003) e Costa *et al.* (2007) aplicaram no pós-processamento para avaliar outros classificadores hierárquicos a fim de corrigir as inconsistências. Mais tarde, Paes, Plastino and Freitas (2012) também utiliza esta técnica como pós-processamento de duas formas diferentes para mitigar as inconsistências. Ao compará-las com o classificador plano e LCPN, pôde-se obter uma boa performance na predição.

Cerri, Barros and Carvalho (2015) propõem o método de rede neural baseado em perceptron multicamadas chamado de *Hierarchical Multi-Label Classification with Local Multi-Layer Perceptrons* (HMC-LMLP). Desta forma, é criado um conjunto de redes neurais, nas quais cada uma é responsável por prever as categorias em um determinado nível. Já Tavares (2018) utiliza classificação por nível do domínio da pergunta escrita pelo usuário para roteamento e seleção de sistemas de *Question Answering* utilizando NB.

Zheng and Zhao (2020) desenvolveram o método *Cost-Sensitive Hierarchical Classification with Imbalanced Classes* (CSHCIC), uma abordagem específica para classificação hierárquica em cenários com classes desbalanceadas. O método implementa uma estratégia de classificação por níveis, utilizando aprendizagem sensível a custos. A metodologia emprega regressão logística com limiares (*thresholds*) específicos para cada nó hierár-

quico, permitindo a mitigação da propagação de erros entre níveis, através da análise da significância das probabilidades estimadas.

Wang *et al.* (2022b) propõem uma abordagem chamada de *Hierarchy-aware Prompt Tuning* (HPT), na qual utiliza um modelo de linguagem pré-treinado (*BERT* (DEVLIN *et al.*, 2019)) e ajuste de *prompt* para classificação. Além disso, a estratégia utilizada é feita por camadas, assim incorporando as informações das categorias da hierarquia.

2.7.3 Classificação Global

Global Classifier (GC), também conhecido como *big-bang*, é uma abordagem que consiste em treinar apenas um único algoritmo para toda a hierarquia uma única vez, conforme ilustrado na Figura 17. A principal vantagem desta metodologia é o uso de apenas um único classificador, em vez de vários, comparado com as abordagens anteriores. Consequentemente, por usar toda a hierarquia para treinamento, não há problemas com a propagação de inconsistências. Contudo, possui desvantagens, como: alta complexidade computacional na fase de treinamento e dificuldade de classificar corretamente novas categorias que não estavam presentes na fase de treinamento (NAIK; RANGWALA, 2018).

Existe uma lacuna na definição precisa entre métodos globais e não globais de CH na comunidade científica. Silla and Freitas (2011) define, por meio da exclusão, que qualquer algoritmo que não seja plano ou local é uma abordagem global. Outros autores definem como global metodologias que possuem as seguintes características:

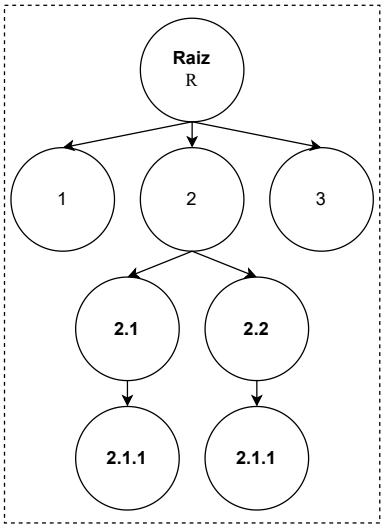
- Utilizam um único algoritmo para treinar toda a hierarquia de classificação de uma só vez;
- Não utilizam informações contextuais locais ou modularização para treinamento do algoritmo;

Há uma pequena quantidade de trabalhos publicados em relação à abordagem global, devido à complexidade para a elaboração de algoritmos globais. Nas subseções a seguir, são apresentados alguns dos métodos desenvolvidos: *Predictive clustering trees* (Seção 2.7.3.1), *Naive Bayes* (Seção 2.7.3.2), *Kernel machines* (Seção 2.7.3.3), Penalização hierárquica (Seção 2.7.3.4) e Redes neurais (Seção 2.7.3.5).

2.7.3.1 *Predictive clustering trees*

Métodos baseados em árvore de decisão na classificação hierárquica são baseados em algoritmos de árvore de decisão plana. BLOCKEEL *et al.* (2002), BLOCKEEL *et al.* (2006) e VENS *et al.* (2008) propõem uma técnica chamada de *Hierarchical multi-label classification with Predictive Clustering Trees (Clus-HMC)*, que desenvolve um conjunto de árvores de classificação para prever todas as classes da hierarquia uma única vez, utilizando

Figura 17 – *Global Classifier* (GC)



Fonte: Silla and Freitas (2011)

como métrica a Distância Euclidiana ponderada para identificar semelhanças de forma mais rápida e não sensíveis à *overfitting*.

Um estudo subsequente explorou o potencial do *Clus-HMC* com variadas medidas de distância (*Jaccard*, *SimGIC* e *ImageClef*), concluindo que não há diferenças significativas entre elas (ALEKSOVSKI; KOCEV; DŽEROSKI, 2009). Perdih *et al.* (2017) propõem utilizar o agrupamento de árvores para classificação da hierarquia de maneira sobreposta, na qual cada nó interno da árvore possui vários agrupamentos hierárquicos alternativos em vez de um único, como proposto anteriormente.

Santos *et al.* (2020) apresentam um estudo para lidar com problemas de classificação multirrótulo utilizando *Predictive Bi-clustering Trees*. Esta abordagem possui a capacidade de prever interações entre dois conjuntos de nós em uma rede bipartida, tanto horizontal quanto verticalmente, em um único conjunto de árvores.

2.7.3.2 *Naive Bayes*

Silla and Freitas (2009) propôs o método probabilístico chamado *Global Model Naive Bayes (GMNB)* aplicado a dados textuais de bioinformática (funções protéicas). Esta abordagem utiliza um único algoritmo que contabiliza as classes atuais e ancestrais para calcular a probabilidade de um exemplo pertencer a uma determinada classe. Graças à adaptação do *Naive Bayes*, o processo de decisão é claramente compreensível e capaz de lidar com dados ausentes.

Diante da eficácia da metodologia GMNB em lidar com dados faltantes em hierarquias, estudos subsequentes desenvolveram o método *Hierarchical Supervised Imputation Method* (HSIM), que emprega algoritmos de GMNB na etapa de pré-processamento (GALVAO; MERSCHMANN, 2016). Lima *et al.* (2018) propõem a técnica *Variable Neighborhood*

Search (VNS), denominada VNS-FSHS, na qual visa selecionar dados relevantes para aplicação no algoritmo GMNB em problemas de CH.

Posteriormente, Lima *et al.* (2021) apresentam um método de seleção híbrida de dados relevantes. Este processo utiliza uma variação de VNS chamada de *General Variable Neighborhood Search* (GVNS), também chamada de GVNS-FSHC, combinada a uma etapa de filtragem por meio de ranqueamento dos dados relevantes com base na estrutura da hierarquia. Após a etapa de pré-processamento, os dados mais relevantes são inseridos no algoritmo GMNB para classificação dos rótulos hierárquicos.

2.7.3.3 *Kernel machines*

Dekel, Keshet and Singer (2004) propôs a combinação de dois métodos: análise *Bayesiana* e *Large Margin Kernels* (VAPNIK, 1999) aplicada à classificação de páginas web. Cai and Hofmann (2004) propõem uma modificação do SVM, chamada de *Hierarchical-SVM* ou HSVM, que considera as relações entre as classes com o objetivo de separar corretamente os caminhos corretos dos incorretos da hierarquia aplicada a uma base de patentes.

Em seguida, uma variação da estrutura *Maximum Margin Markov Network* chamada de *kernel-based method* foi proposta por ROUSU *et al.* (2005) e ROUSU *et al.* (2006) para resolver um problema de CH multirrótulo, decompondo-a em subproblemas contendo uma única amostra e o emprego do gradiente antecedente condicional para a otimização das predições. Entretanto, tais estudos ficaram restritos a pequenas bases de dados, sendo pouco escaláveis para problemas mais complexos.

Outro *kernel-based method* foi apresentado por Seeger (2008) propondo uma abordagem mais eficiente aplicável a grandes bases de dados com rótulos multiclasse. Diferente dos estudos anteriores, este método buscou a otimização de *Newton*, que consiste no uso da estrutura do modelo e gradientes lineares conjugados para encontrar de forma mais rápida e aproximada as direções de *Newton* e, consequentemente, a classificação dos exemplos de toda a hierarquia uma única vez. Seeger (2008) afirmam que esta abordagem é mais generalizável e facilmente aplicada a outros problemas, sem necessidade de realizar muitas alterações.

Qiu, Gao and Huang (2009) propôs uma nova abordagem chamada de *Global Margin Maximization* (HSVM-S), na qual separa corretamente todos os nós de seus irmãos da hierarquia e, em seguida, os incorpora como informações para maximizar as margens de todas as categorias da hierarquia.

2.7.3.4 Penalização hierárquica

Outra abordagem explorada na literatura utiliza penalizações hierárquicas para lidar com problemas de CH. Gopal and Yang (2013) utiliza *Binary Cross-Entropy* (BCE)

ou *log loss*, como função de custo, que penaliza classificações incorretas das classes da hierarquia, incorporando as relações hierárquicas na regularização dos parâmetros do modelo. Desta forma, classes mais próximas na hierarquia compartilham parâmetros semelhantes.

Em seguida, Zhang *et al.* (2021) apresentaram outra abordagem que utiliza a regularização dos parâmetros baseada em hiperonímia. Hiperonímia é a relação semântica que estabelece uma hierarquia conceitual, na qual uma palavra de significado mais amplo (hiperonímia) engloba termos mais específicos (hiponímia). Assim, os exemplos que possuem palavras que compartilham o mesmo hiperônimo apresentam maior semelhança. Ao serem inseridos na regularização dos parâmetros, auxiliam o modelo na compreensão das relações desses exemplos dentro das classes na hierarquia, principalmente nas relações entre os nós pai e filho.

Posteriormente, Vaswani *et al.* (2022) apresentaram uma nova função de custo chamada de *Comprehensive Hierarchy Aware Multi-label Predictions* (CHAMP), que é aplicável a problemas de CH multirrótulo. O método incorpora um cálculo de erro hierárquico, que avalia a distância entre a predição e a verdade fundamental na árvore de hierarquia, para determinar a penalidade proporcional à gravidade do erro.

2.7.3.5 Redes neurais

Estudos mais recentes têm explorado redes neurais como possibilidade de interpretar a estrutura hierárquica e na tarefa de classificação dos rótulos. Alguns trabalhos propõem incorporar as informações das categorias como: aprendizagem por reforço (MAO *et al.*, 2019) e redes encapsuladas (ALY; REMUS; BIEMANN, 2019).

Zhou *et al.* (2020) propõem incorporar a combinação de duas informações: sendo elas as categorias, vindas das informações da estrutura hierárquica, e informações do texto. Através deste estudo, diversas pesquisas foram desenvolvidas com o objetivo de entender como ambas as informações poderiam ser utilizadas em conjunto para obter melhores performances. CHEN *et al.* (2020) e CHEN *et al.* (2021) abordam o problema de CH através da relação semântica do texto e rótulo da hierarquia, incorporando-os juntos para transformá-los em um problema de combinação semântica. Deng *et al.* (2021) maximiza as informações comuns entre os rótulos, com o objetivo de restringir o aprendizado dos rótulos. Zhao *et al.* (2021) extrai informações dos textos e rótulos através da fusão de autoadaptação.

Wang *et al.* (2022a) propõem uma abordagem de aprendizagem contrastiva, que consiste em inserir informações da hierarquia nas informações textuais, auxiliando na seleção dos *tokens* que de fato se relacionam com os rótulos. Essas informações são inseridas como amostras para realizar a aprendizagem contrastiva. Zhang *et al.* (2022) utiliza o módulo de atenção nas redes neurais para incorporação das informações das categorias.

Já Zhu *et al.* (2023) desenvolveu uma abordagem que transforma a hierarquia através da entropia estrutural como informação para treinamento do modelo. Recentemente, Zhu *et al.* (2024) combinou ambas as ideias.

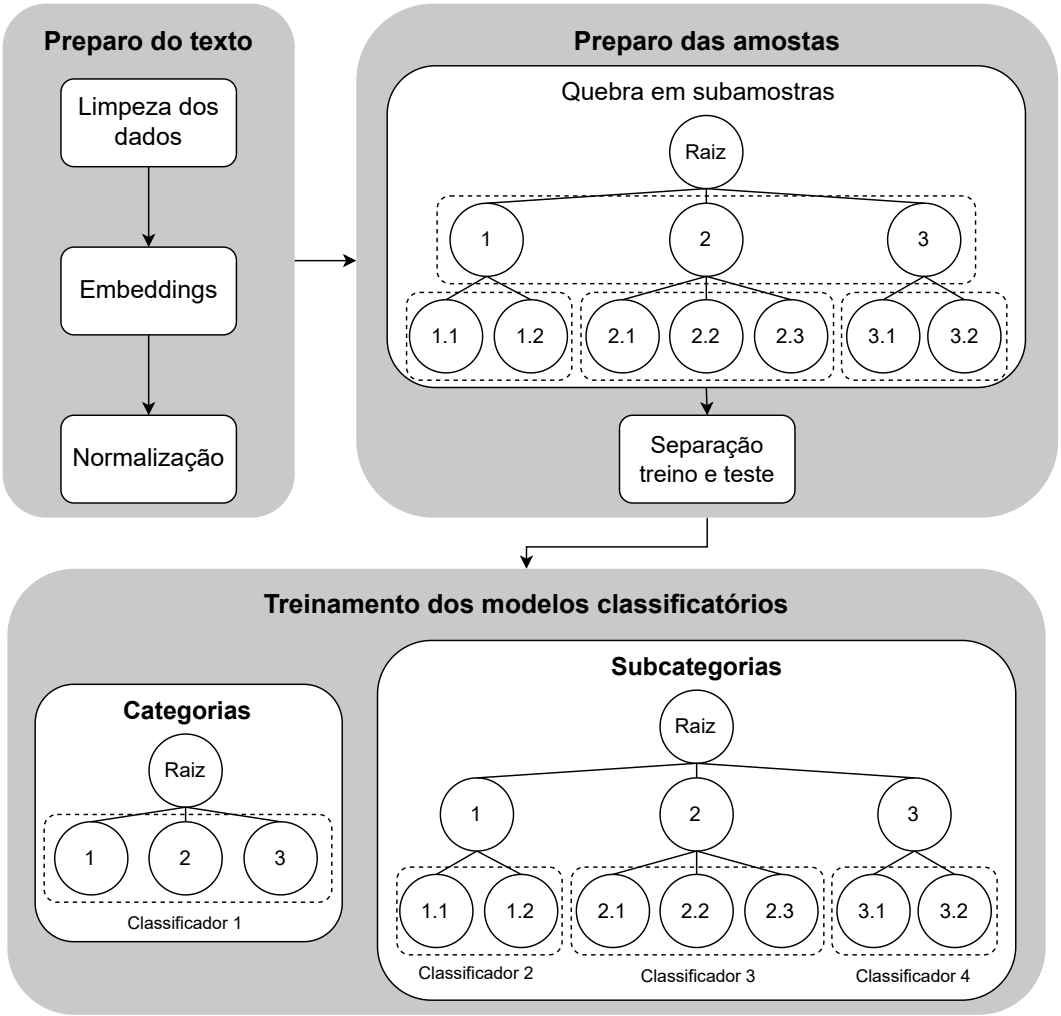
2.8 Considerações finais

Este capítulo introduziu os principais fundamentos relacionados à CH de textos sob a perspectiva do AM. Aprofundou-se nas técnicas mais tradicionais de pré-processamento e representação computacional de textos, descrevendo os desafios inerentes à transformação de dados não estruturados em representações vetoriais, como, por exemplo, a captura do sentido semântico da linguagem. Na discussão sobre algoritmos supervisionados, foram apresentados os tipos de classificação (binária, multiclasse, multirótulo e hierárquica), culminando com um foco especial na CH. A apresentação detalhada das definições, estruturas e profundidades hierárquicas demonstrou a complexidade e as nuances envolvidas nesta abordagem específica de classificação. Com o objetivo de guiar o leitor nas seções subsequentes, foi apresentada uma análise dos diferentes tipos de métodos de CH e os principais trabalhos relacionados.

3 PROPOSTA

O presente trabalho tem como objetivo realizar a classificação hierárquica de textos de chamados, estruturando o processo em etapas específicas ilustradas na Figura 18. Inicialmente, os textos passam por um processo de pré-processamento, no qual são convertidos em representações numéricas (*embeddings*), facilitando sua interpretação pelos modelos de AM. Em seguida, os dados são divididos de acordo com a hierarquia das classes, criando subconjuntos específicos (*subdatasets*) para cada nível hierárquico. Esses subconjuntos são então divididos em conjuntos de treino e teste, permitindo que os algoritmos sejam treinados separadamente em cada nível hierárquico; esses passos serão melhor detalhados na Seção 3.1. Posteriormente, os modelos são avaliados quanto ao seu desempenho, garantindo a validação e análise de sua eficácia no contexto proposto; essa etapa será discutida na Seção 3.2.

Figura 18 – Fluxo da proposta



Fonte: Autor

3.1 Preparação do texto e amostras

O pré-processamento dos textos é uma etapa essencial para garantir que os dados estejam em um formato adequado para a aplicação dos algoritmos de AM. Essa fase permite transformar os textos brutos em representações estruturadas e compreensíveis para os modelos, o que é fundamental para melhorar o desempenho na tarefa de classificação conforme foi discutido previamente na Seção 2.2.

A etapa de limpeza dos textos é opcional e sua necessidade depende diretamente das características dos dados utilizados. O objetivo principal dessa etapa é remover informações que não sejam relevantes ou que possam prejudicar as etapas subsequentes. Por exemplo, em abordagens que utilizam *Bag-of-Words* (BoW) como método de representação textual, é necessário incluir neste processo todos os pré-processamentos relevantes para garantir uma representação adequada, como a remoção de *stopwords*, normalização de palavras, entre outros. No caso deste trabalho, que foca no uso de *embeddings* para a representação dos textos, a limpeza se limita à remoção de elementos que possam ser irrelevantes ou prejudiciais ao contexto, como links, *emojis* e *hashtags*, dependendo de sua relevância nos dados utilizados.

Os textos podem ser representados utilizando qualquer uma das abordagens discutidas na Seção 2.2, como BoW, TF-IDF ou *embeddings*. Entretanto, o foco deste trabalho recai sobre o uso de *embeddings*, que oferece uma representação densa e rica em informações semânticas, capturando relações contextuais entre palavras de maneira mais eficaz. A escolha por *embeddings* justifica-se pela sua capacidade de generalização, especialmente em problemas de classificação hierárquica, onde a relação semântica é crucial. Nesse processo, o tokenizador associado ao modelo de *embeddings* desempenha um papel fundamental, pois é responsável por dividir os textos em subpalavras ou *tokens* de forma a garantir uma representação adequada aos modelos pré-treinados utilizados.

A etapa de normalização dos dados foi incorporada na metodologia, respaldada por estudos recentes que evidenciam a influência positiva de determinados tipos de normalização no desempenho de modelos de classificação (AHMED *et al.*, 2022; SUJON *et al.*, 2024).

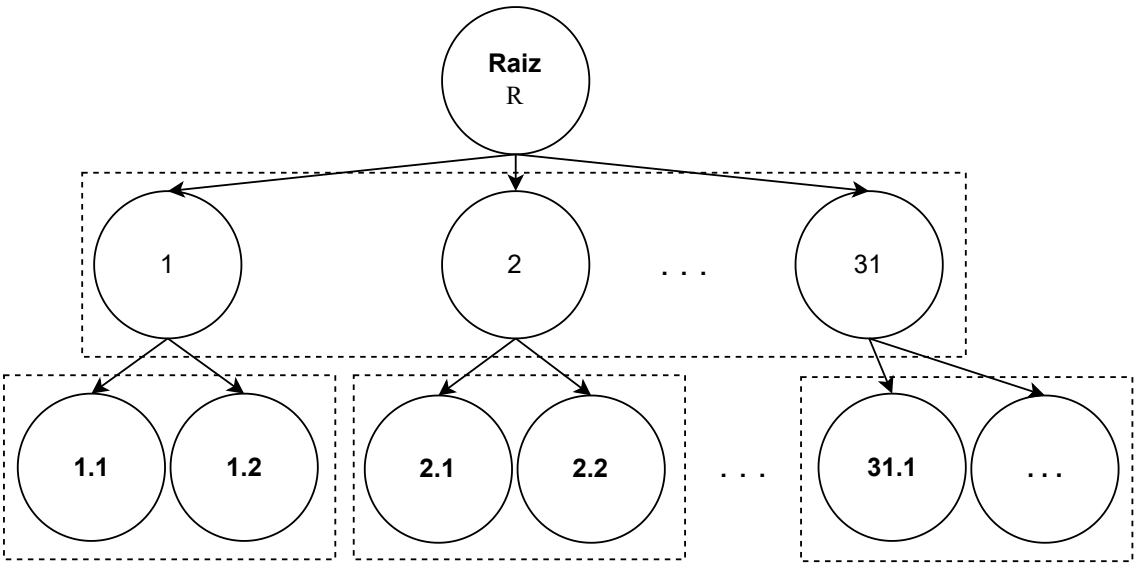
Para organizar os dados de forma a atender à classificação hierárquica, a base de dados foi dividida seguindo a hierarquia das classes. Nesse processo, cada vértice pai na estrutura hierárquica é transformado em um *subdataset*, onde os vértices filhos correspondem às classes desse *subdataset*. Essa abordagem, conhecida como abordagem de irmãos, foi detalhada na Seção 2.7.2.1 e está ilustrada na Figura 13. A escolha dessa metodologia justifica-se por sua simplicidade e eficiência na modelagem de hierarquias complexas, permitindo que os modelos sejam treinados de forma independente em cada nível hierárquico. Além disso, essa estratégia reduz a complexidade do problema, uma vez que os modelos lidam apenas com os relacionamentos locais entre os nós, em vez de toda

a hierarquia, o que facilita a interpretação dos resultados e a detecção de possíveis erros em níveis específicos da estrutura.

3.2 Treinamento

A metodologia de treinamento implementada neste trabalho fundamenta-se em uma adaptação da abordagem LCL. Esta variação se caracteriza pela utilização de classificadores específicos para cada conjunto de nós que compartilham o mesmo nó pai, conforme ilustrado na Figura 19.

Figura 19 – Configuração dos classificadores no modelo proposto



Fonte: Metz (2011)

O processo de classificação durante a fase de teste segue uma estrutura hierárquica, iniciando-se no nível mais superficial da árvore. A predição obtida neste nível determina qual classificador será acionado no nível subsequente. Esta abordagem incorpora duas características fundamentais: é multiclasse, uma vez que seleciona uma única opção dentre múltiplas classes disponíveis, e MLNP, pois requer obrigatoriamente a seleção de um nó-folha.

A abordagem escolhida, conforme ilustrado na Figura 19, oferece soluções para diversos desafios encontrados nas abordagens tradicionais de classificadores locais (LCN, LCPN e LCL). Uma vantagem significativa reside na resolução de inconsistências em tarefas multiclasse, dado que as predições realizadas nos níveis mais profundos mantêm coerência com as decisões tomadas nos níveis superiores.

Esta consistência é garantida pela estrutura única da subárvore associada a cada nó selecionado, eliminando a possibilidade de atribuição de classes desconexas à mesma amostra. Por exemplo, conforme ilustrado na Figura 19, se o nó 1 é selecionado, o processo de classificação prossegue exclusivamente para sua subárvore, contemplando apenas seus

nós-filho (1.1, 1.2). Consequentemente, nós pertencentes a outras subárvores, como aqueles sob o nó 2 a 31, são automaticamente excluídos do processo de seleção.

Além disso, esta abordagem apresenta vantagens operacionais significativas em relação às abordagens tradicionais. Uma característica notável é a redução substancial no número total de classificadores necessários quando comparada à abordagem LCN, o que resulta em maior eficiência computacional. Ademais, a arquitetura modular do sistema proporciona flexibilidade significativa na manutenção e evolução do modelo. Esta modularidade permite tanto a incorporação de novos domínios quanto a substituição seletiva de classificadores que apresentem deterioração em seu desempenho, mantendo intactos aqueles que continuam operando com eficácia satisfatória. Tal característica representa uma vantagem considerável em relação à abordagem global, onde modificações frequentemente requerem o retreinamento completo do sistema, resultando em maior complexidade operacional e consumo de recursos computacionais.

3.3 Considerações finais

A proposta metodológica apresentada neste capítulo integra duas etapas fundamentais: um robusto *pipeline* de pré-processamento textual e uma abordagem hierárquica para o treinamento do modelo. Na primeira etapa, a utilização de modelos de linguagem oferece vantagens significativas, especialmente naqueles baseados na arquitetura *Transformer* e que sejam multilíngues, permitindo uma representação mais rica e contextualizada dos textos em português. A normalização dos dados é uma etapa opcional no pré-processamento, pois, segundo diversos estudos, pode ou não otimizar o desempenho, dependendo do classificador escolhido.

Na etapa de treinamento, a adaptação proposta da abordagem LCL apresenta uma solução elegante para os desafios intrínsecos à classificação hierárquica de textos. Esta abordagem não apenas mantém a consistência nas predições através dos diferentes níveis da hierarquia, mas também oferece vantagens práticas como a redução do número de classificadores necessários e maior facilidade de extensão para novos domínios.

A integração destas duas etapas resulta em um *framework* metodológico que busca maximizar a precisão na classificação hierárquica de textos, enquanto mantém a flexibilidade necessária para adaptações e extensões futuras. Esta abordagem representa um equilíbrio entre robustez técnica e praticidade de implementação, características essenciais para sua aplicação efetiva em cenários reais de classificação textual.

4 AVALIAÇÃO EXPERIMENTAL

Este capítulo apresenta a análise e os resultados das avaliações experimentais conduzidas para validar as propostas descritas no Capítulo 3. A Seção 4.1 detalha as características fundamentais do conjunto de dados empregado nos experimentos. A metodologia e as configurações experimentais são descritas na Seção 4.2, seguidas pela apresentação das métricas de avaliação utilizadas na Seção 4.3. Por fim, a Seção 4.4 apresenta e discute os resultados obtidos.

4.1 Conjuntos de Dados

Foram conduzidos experimentos utilizando uma base de dados que contém a descrição do problema que o cliente possui e a classificação da categoria e subcategoria do motivo do chamado, feita pelo suporte ao fim do atendimento. O conjunto de dados em questão contém 226.438 exemplos reais destas anotações organizados em 31 categorias e 240 subcategorias. As categorias representam os principais temas dos chamados e as subcategorias especificam os problemas ou questões dentro de cada categoria.

Os assuntos dos atendimentos podem ser tratados como um problema multirrótulo, visto que podem existir múltiplas categorias e subcategorias atreladas a um único chamado. Por exemplo, se o cliente estiver com problemas para emissão de uma nota fiscal, pode acontecer de o problema estar atrelado ao cadastro da venda que foi feito de maneira incorreta, impedindo-o de realizar esta operação. Desta forma, o chamado pode ser classificado, ao mesmo tempo, nas categorias “Emissão de NF-e” e “Cadastro de vendas”; e nas subcategorias “Falha na emissão de NF-e” e “Frente de caixa”, por exemplo (conforme ilustrado na Tabela 3).

Tabela 3 – Exemplos temas de chamados

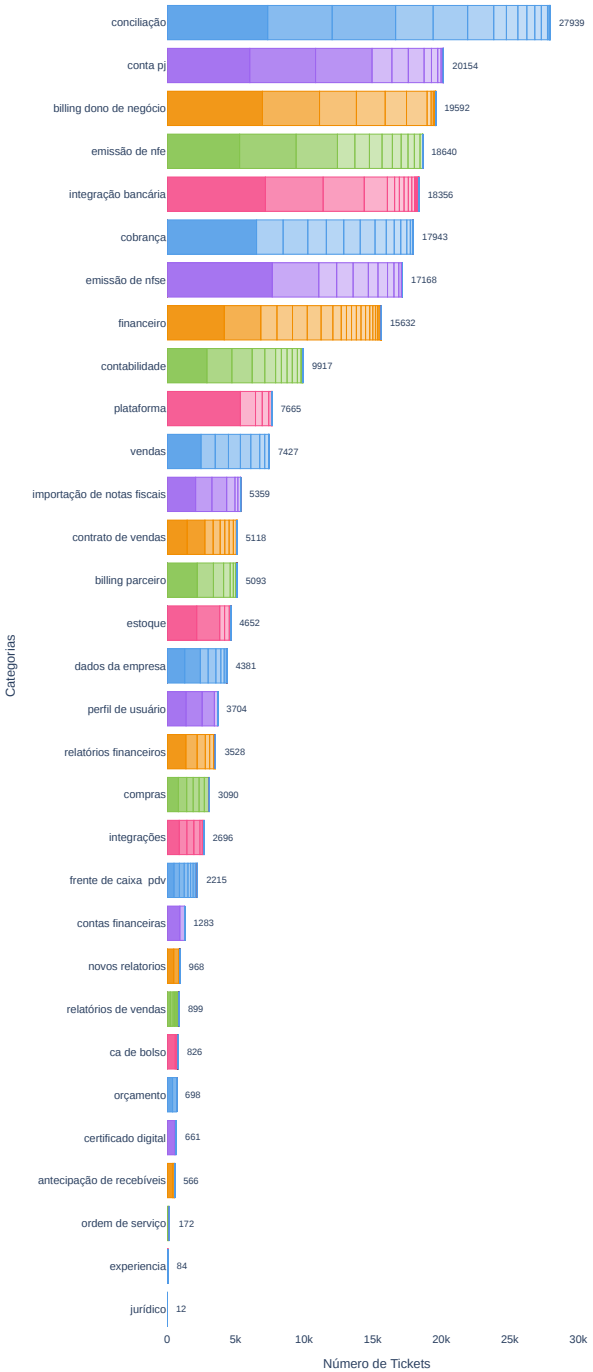
Categoria	**Subcategoria**
Emissão de NF-e	Como emitir NF-e **Falha na emissão de NF-e** Cancelamento e inutilização de NF-e
Cadastro de vendas	**Frente de caixa** *E-commerce*

Entretando, para fins de delimitação inicial do escopo experimental, o presente estudo utilizou uma base de dados constituída exclusivamente por exemplos multi-classe, onde cada instância está associada a uma única classe em cada nível da hierarquia.

Esta decisão metodológica foi adotada para permitir uma análise mais controlada do comportamento do modelo proposto.

Outra característica importante da base de dados é um grande desbalanceamento entre as classes, impactando diretamente na performance dos modelos. Conforme ilustrado na Figura 20, as categorias do primeiro nível possuem grande diferença na distribuição dos tickets. Além disso, das 240 subcategorias, 71 delas possuem menos de 10 exemplos para serem separados em treino e teste (o que representa 30% de toda a base).

Figura 20 – Distribuição dos tickets por Categoria e Subcategoria



Fonte: Autor

4.2 Configuração Experimental

Conforme previamente detalhado na Seção 3.1, a transformação dos dados textuais em *embeddings* constitui uma etapa fundamental do processo, viabilizando uma interpretação mais eficiente pelos algoritmos de AM. Para esta finalidade, foi selecionado o modelo de linguagem *Multilingual E5 Text Embeddings* (WANG *et al.*, 2024), essa escolha foi motivada por diversas características técnicas relevantes.

A arquitetura do modelo de linguagem escolhido é baseada em *Transformer* e oferece duas vantagens significativas. Primeiramente, a família de modelos E5 foi treinada especificamente para geração de *embeddings* de qualidade. Adicionalmente, seu treinamento com extensos conjuntos de dados multilíngues resulta em uma compreensão mais robusta e abrangente da linguagem. O *Multilingual E5 Text Embeddings*, em particular, foi especializado em 16 idiomas diferentes, contemplando o português, e sua versão *large* incorpora 24 camadas e uma representação contextual de dimensão 1024, o que permite uma codificação mais rica do contexto em comparação com versões de dimensionalidade inferior.

Na implementação dos classificadores, adotou-se *Naive Bayes* (NB) e *Multilayer Perceptron* (MLP) com duas variações arquiteturais: MLP(256, 128) e MLP(512). Esta escolha metodológica do NB fundamenta-se pelo menor custo computacional e para MLP nas características intrínsecas da classificação textual, que se distingue pela alta dimensionalidade inerente às *embeddings* de contexto, cuja representação complexa demanda arquiteturas de redes neurais capazes de capturar nuances semânticas.

Para maximizar o desempenho do classificador NB, foram exploradas as seguintes configurações: o parâmetro *alpha*, que controla a suavização de *Laplace*, foi avaliado com os valores 0.1, 0.5, 1.0 e 1.5, permitindo ajustar o nível de regularização do modelo. Adicionalmente, o parâmetro *fit prior*, responsável por determinar se as probabilidades das classes devem ser consideradas durante o aprendizado, foi testado em suas duas configurações possíveis (verdadeiro e falso).

Já para a otimização do desempenho computacional e robustez dos classificares MLP, foi estabelecido 1000 épocas, que visam garantir que o modelo tenha tempo suficiente para convergência, permitindo que os algoritmos de aprendizado explorem exaustivamente os padrões nos dados sem incorrer em sobreajustamento prematuro. A implementação da técnica de *dropout* entre camadas constitui uma estratégia fundamental para mitigar o sobreajustamento, reduzindo a dependência excessiva do modelo em características específicas do conjunto de treinamento e promovendo maior capacidade de generalização.

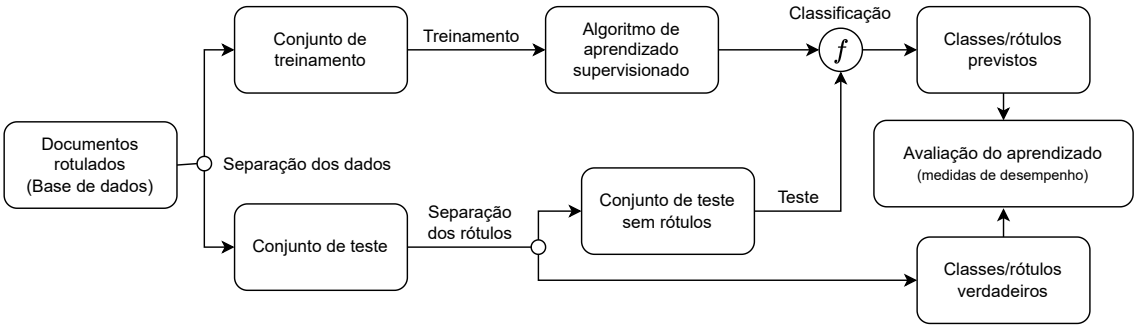
A seleção do $F1_{macro}$ como métrica de avaliação fundamenta-se em sua robustez para lidar com conjuntos de dados desbalanceados, proporcionando uma representação mais equitativa do desempenho do modelo dentre as diferentes classes. A utilização da função

de perda *categorical crossentropy* mostra-se particularmente adequada para problemas de classificação multiclasse, permitindo uma estimativa precisa da divergência entre as distribuições de probabilidade preditas e reais. O otimizador *Adam* foi escolhido por sua eficácia comprovada em adaptar taxas de aprendizado para cada parâmetro, facilitando a convergência em problemas de aprendizado complexos. Por fim, o *Early Stopping*, configurado com um intervalo de 10 épocas, representa uma estratégia de regularização que interrompe o treinamento quando não se observa melhoria significativa, prevenindo o desperdício computacional e mitigando potenciais riscos de sobreajustamento.

4.3 Métricas de Avaliação

Para avaliar a eficácia de um modelo de classificação, um conjunto de documentos rotulados é separado como um conjunto de teste e não é utilizado durante o treinamento. Os documentos no conjunto de teste são então classificados utilizando o modelo de classificação já treinado, e os rótulos previstos são comparados com os rótulos verdadeiros, conforme ilustrado na Figura 21. Desta forma, é possível calcular medidas de desempenho a partir dessa comparação.

Figura 21 – Etapas de criação de um modelo supervisionado



Fonte: Autor

Existem diversas medidas de desempenho que podem ser analisadas como: acurácia dos resultados, tempo de aprendizado, qualidade do conhecimento extraído, entre outros. Nesta seção, serão apresentadas as principais medidas de avaliação de desempenho de algoritmos de classificação utilizadas para avaliar o experimento. Serão apresentadas as medidas para classificação binária, mas elas podem ser utilizadas para outros tipos de classificação, dependendo da métrica. Inicialmente, define-se a classe positiva como (+) e a classe negativa como (-), assim sendo possível construir a Matriz de Confusão que possui os seguintes valores:

- Verdadeiros Positivos (VP): valor que correponde ao total de números de documentos da classe positiva classificados corretamente;

- Verdadeiros Negativos (VN): valor que correponde ao total de números de documentos da classe negativa classificados corretamente;

- Falsos Positivos (FP): valor que correponde ao total de números de documentos da classe positiva classificados incorretamente;

- Falsos Negativos (FN): valor que correponde ao total de números de documentos da classe negativa classificados incorretamente.

A partir destes valores, temos que $n = VP + VN + FP + FN$, é possível criar outras medidas de desempenho:

- Precisão (em inglês, *precision*): proporção de documentos positivos classificados corretamente considerando o total de documentos preditos como positivos (conforme apresentado na Equação 4.1). Valores altos de precisão como, por exemplo, 1 indicam que o modelo classifica corretamente exemplos que são da classe positiva, mas desconsidera aqueles exemplos que são da classe positiva e não foram devidamente classificados.

$$prec(\hat{f}) = \frac{VP}{VP + FP} \tag{4.1}$$

- Sensibilidade ou revocação (em inglês, *sensitivity* ou *recall*, também chamado de taxa de verdadeiros positivos - TVP): proporção de documentos da classe positiva classificados corretamente (conforme apresentado na Equação 4.2). Valores altos de revocação como, por exemplo, 1 indicam que o modelo classifica corretamente exemplos que são da classe positiva, considerando os exemplos que são da classe positiva e não foram devidamente classificados.

$$sens(\hat{f}) = rev(\hat{f}) = TVP(\hat{f}) = \frac{VP}{VP + FN} \tag{4.2}$$

- Especificidade (em inglês, *specificity*, seu complemento corresponde à taxa TFP): proporção de documentos da classe negativa classificados corretamente (conforme apresentado na Equação 4.3). Valores altos de especificidade como, por exemplo, 1 indicam que o modelo classifica corretamente exemplos que são da classe negativa, mas desconsidera aqueles exemplos que são da classe negativa e não foram devidamente classificados como positivos.

$$esp(\hat{f}) = \frac{VN}{VN + FP} = 1 - TFP(\hat{f}) \tag{4.3}$$

- Medida-F (em inglês, *F-measure*), média harmônica ponderada da precisão e a revocação (conforme apresentado na Equação 4.4).

$$F_w(\hat{f}) = \frac{(w+1) \times rev(\hat{f}) \times prec(\hat{f})}{rev(\hat{f}) + w \times prec(\hat{f})} \tag{4.4}$$

Em contraste com métricas anteriores, que frequentemente apresentam viés para classes específicas (positivas ou negativas) e susceptibilidade ao desbalanceamento dos dados, a Medida-F caracteriza-se pela capacidade de atribuir pesos equivalentes $w = 1$ entre todas as classes. Esta ponderação igualitária confere o mesmo grau de importância para revocação e precisão, resultando na métrica $F1_{macro}$, conforme ilustrado na Equação 4.5:

$$F_1(\hat{f}) = \frac{2 \times rev(\hat{f}) \times prec(\hat{f})}{rev(\hat{f}) + prec(\hat{f})} \tag{4.5}$$

Esta propriedade a torna particularmente adequada para a avaliação de modelos em contextos de classificação multiclasse e multirrótulo, especialmente em cenários onde há significativo desbalanceamento na distribuição das classes.

4.4 Resultados e Discussões

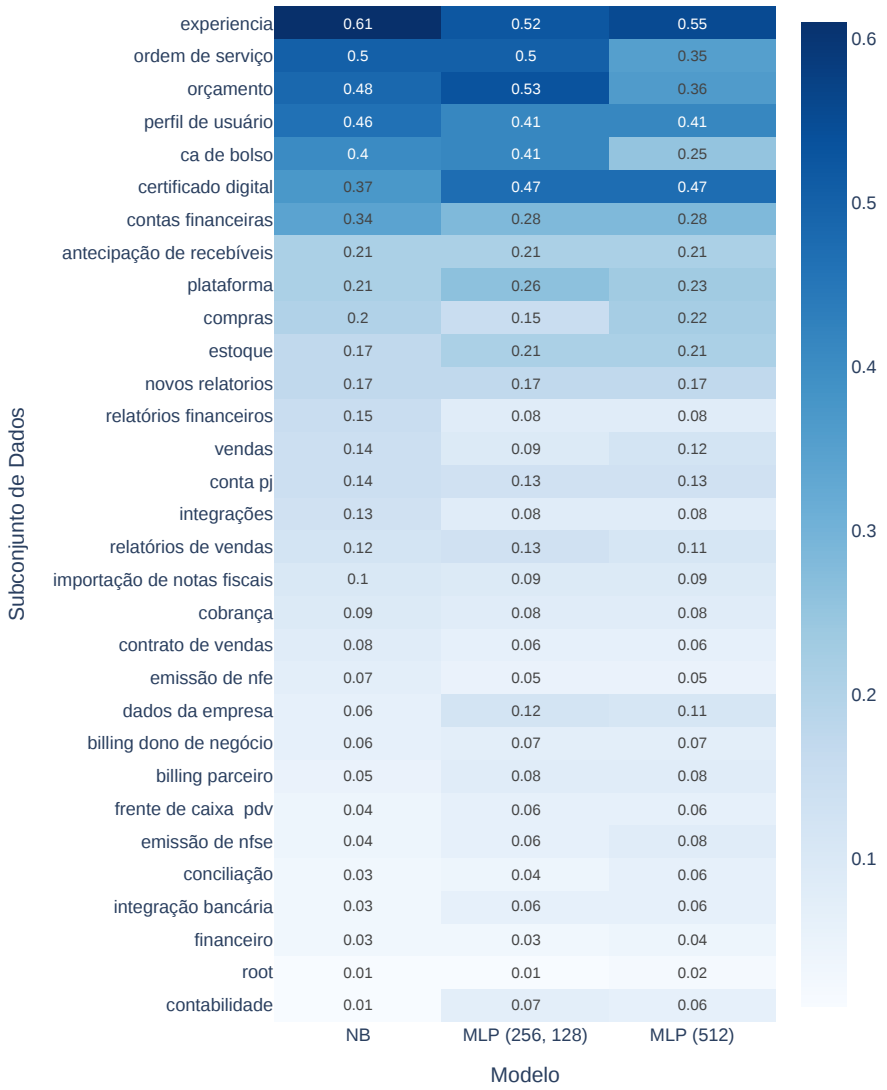
Conforme ilustrado na Figura 22, as médias de $F1_{macro}$ obtidas nas execuções de cada classificador para cada conjunto de dados, apresentadas em um mapa de calor com gradientes de azul, no qual tonalidades mais escuras representam desempenhos superiores.

Observa-se que todos os classificadores apresentaram valores de $F1_{macro}$ limitados, não ultrapassando 61% em nenhum dos conjuntos de dados, independentemente do nível hierárquico analisado. Apesar das variações na arquitetura das redes neurais multicamadas (MLP(256, 128) e MLP(512)), seus desempenhos revelaram-se notavelmente similares, sugerindo que a configuração de camadas e neurônios não constituiu fator determinante para a discriminação das classes.

A complexidade na classificação pode ser atribuída a múltiplos fatores, sendo o número de classes e subclasses da taxonomia um elemento crítico. Quanto maior a diversidade taxonômica, mais desafiadora torna-se a discriminação entre categorias, especialmente em cenários de distribuição assimétrica de amostras. Além disso, para compreender as limitações de desempenho dos classificadores, realizou-se uma análise mais aprofundada do conjunto de dados, considerando potenciais variáveis intervenientes.

A primeira análise concentrou-se no comprimento das descrições textuais dos tickets, pressupondo que descrições mais detalhadas proporcionariam maior contextualização para o treinamento dos algoritmos. Conforme ilustrado na Figura 23, as descrições dos chamados

Figura 22 – Mapa de calor por Classificador e Método



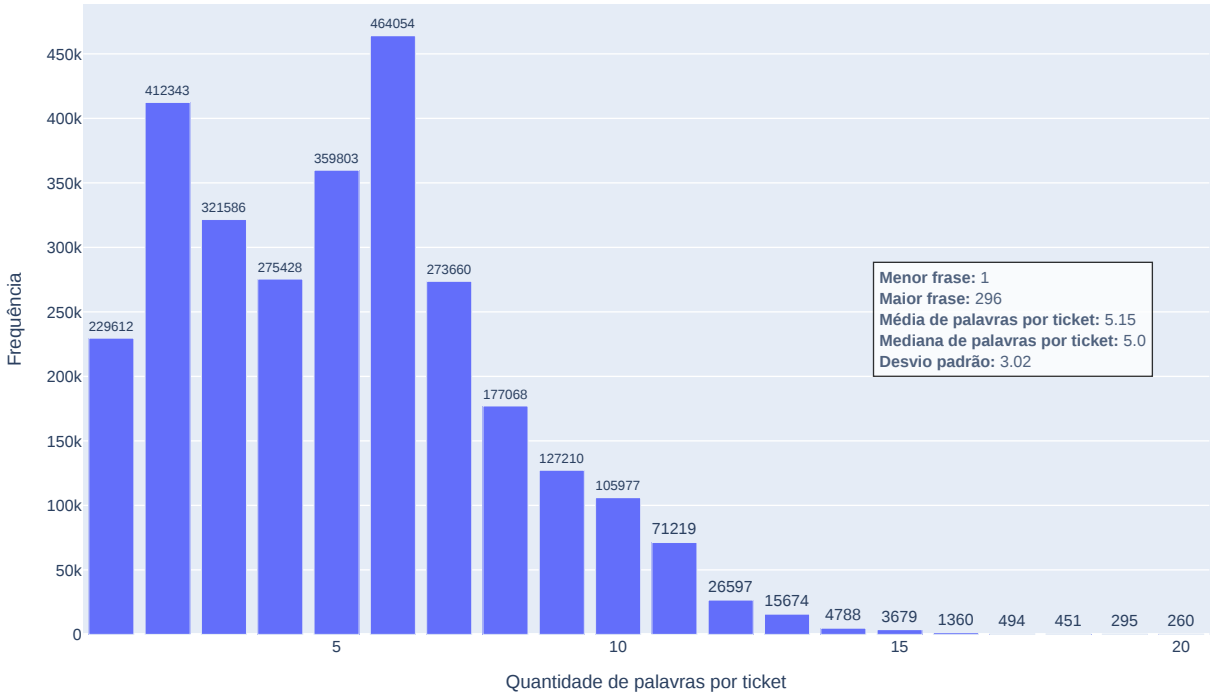
Fonte: Autor

apresentam-se significativamente concisas, resultando em um contexto informacional restrito. Essa limitação textual representa um fator potencialmente limitante para a capacidade discriminativa dos modelos de classificação.

Adicionalmente, investigou-se a similaridade semântica entre as descrições dos chamados, aspecto que pode comprometer a diferenciação entre categorias e subcategorias. A Figura 24, utilizando *embeddings* de uma categoria representativa, elucida a complexidade inerente à segmentação dos tickets em subcategorias. A proximidade semântica entre descrições distintas introduz um desafio significativo para os algoritmos de aprendizado.

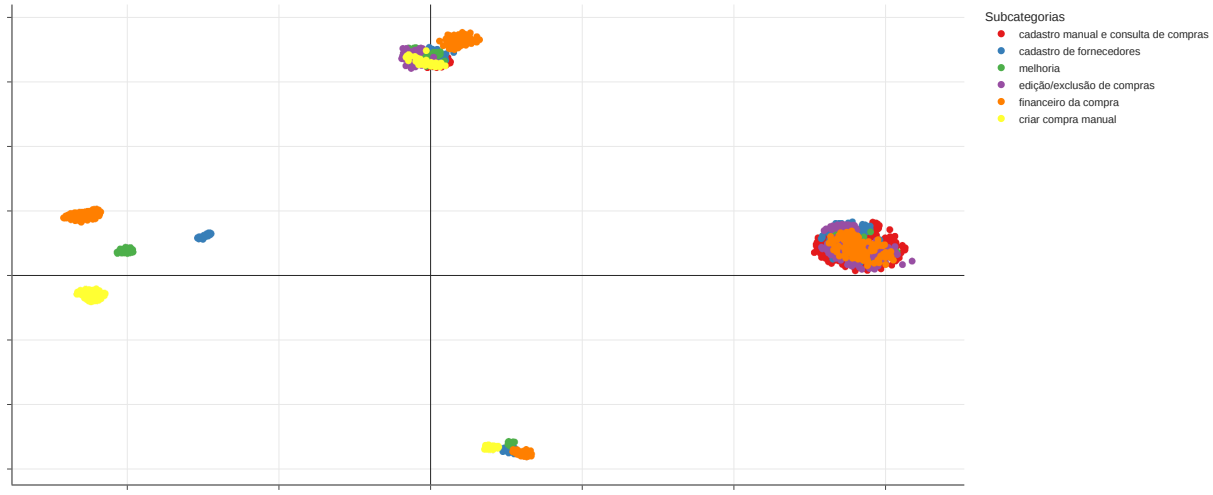
Por fim, o contexto específico do problema envolve terminologias especializadas e jargões pertencentes aos domínios contábil e financeiro, característicos da organização em estudo. Essa particularidade linguística pode ter constituído uma limitação significativa, dado que os modelos de *embeddings* preexistentes provavelmente não foram treinados com vocabulário tão específico. A ausência de representações vetoriais adequadas para esse

Figura 23 – Distribuição da quantidade de palavras por ticket



Fonte: Autor

Figura 24 – Similaridade semântica dos tickets das subcategorias de Compras



Fonte: Autor

léxico especializado pode ter comprometido a capacidade de generalização e discriminação dos algoritmos de classificação.

Em síntese, o processo de aprendizado no contexto de classificação hierárquica é multifacetado e influenciado por diversas variáveis. No experimento em questão, identificaram-se desafios substanciais relacionados à informação textual utilizada no treinamento dos algoritmos, destacando a necessidade de abordagens metodológicas mais sofisticadas para aprimorar a precisão classificatória.

4.5 Considerações finais

Este capítulo apresentou uma descrição detalhada do processo experimental, abrangendo as técnicas metodológicas empregadas, as métricas de avaliação utilizadas e uma análise aprofundada dos conjuntos de dados. Os cenários experimentais foram sistematicamente explorados, permitindo uma análise comparativa dos resultados obtidos.

A partir dos resultados experimentais, conclui-se que a hipótese inicial de automatizar a categorização de chamados utilizando métodos consolidados de classificação hierárquica de textos não foi corroborada. Esta conclusão fundamenta-se em dois aspectos críticos identificados: o significativo desbalanceamento na distribuição das amostras e a forte interrelação entre categorias e subcategorias, que dificulta a separação clara entre as classes. Estas características intrínsecas do conjunto de dados impõem desafios substanciais à aplicação direta dos métodos tradicionais de classificação hierárquica.

5 CONCLUSÕES

A classificação automática de chamados representa uma capacidade estratégica significativa para organizações, possibilitando a tomada de decisões ágil e o acionamento preventivo de áreas específicas em situações críticas, como na identificação de falhas sistêmicas e instabilidades operacionais. Esta automatização também oferece potencial de aplicação em diversos contextos de atendimento ao consumidor, incluindo o aprimoramento de sistemas conversacionais (*chatbots*), a otimização da identificação temática e o enriquecimento de bases de conhecimento, como FAQs e portais de perguntas e respostas.

No entanto, o cenário atual apresenta desafios significativos: o expressivo volume de atendimentos diários inviabiliza a análise manual em tempo real, e a complexidade da estrutura temática dos chamados, organizada em uma taxonomia hierárquica de categorias e subcategorias inter-relacionadas, demanda abordagens sofisticadas de classificação.

Diante deste contexto, o presente trabalho propôs a aplicação de técnicas de CH de textos especificamente direcionadas à categorização automática de chamados de suporte em ambiente organizacional.

O experimento foi projetado tendo como base as principais características de um método ideal para manipular os problemas de CH de texto no contexto atual: uso de LCL, na qual mantém a consistência nas predições através dos diferentes níveis da hierarquia, como também oferece vantagens práticas como a redução do número de classificadores necessários e maior facilidade de extensão para novos domínios.

Para geração das *embeddings*, foi utilizado *Multilingual E5 Text Embeddings*, que possibilita uma representação semântica mais densa e rica, uma vez que captura relações contextuais entre palavras de maneira mais eficaz. Utilizaram-se modelos de *Naive Bayes* e redes neurais multicamadas (MLP(256, 128) e MLP(512)) como classificadores base, cujos resultados experimentais revelaram limitações significativas no processamento de informações textuais especializadas organizadas em diversas categorias e subcategorias desbalanceadas.

A investigação evidenciou que a categorização automática de chamados digitais em linguagem natural configura-se como um desafio complexo, que demanda estratégias metodológicas avançadas para aprimorar a precisão classificatória. Os resultados sublinham a necessidade de abordagens mais refinadas no tratamento de textos com terminologias específicas e estruturas semânticas intrincadas. Como direcionamentos para pesquisas futuras, propõem-se as seguintes extensões e aprimoramentos:

- Desenvolvimento de uma abordagem híbrida utilizando a combinação de BoW

e *embeddings*, possibilitando uma adaptação maior para capturar as nuances do contexto fiscal e contábil brasileiro;

- Investigação comparativa de diferentes arquiteturas de modelos e estruturas hierárquicas, visando identificar configurações mais adequadas ao domínio específico;
- Expansão do escopo para incluir classificação multirrótulo, reconhecendo que instâncias podem pertencer simultaneamente a múltiplas classes, representando assim cenários mais próximos da realidade operacional;
- Com o cenário multirrótulo, é necessário implementar uma nova métrica de avaliação combinada, incorporando o *Hamming Score* normalizado (invertido para maximização) em conjunto com a média harmônica $F1_{macro}$, proporcionando uma avaliação mais abrangente do desempenho do modelo;
- Adoção de validação cruzada utilizando *Stratified K-Fold*, garantindo a preservação da distribuição das classes nos conjuntos de treino e teste, visando uma avaliação mais robusta e confiável do desempenho do modelo;
- Incorporação de um conjunto de dados independente para testes, simulando condições reais de operação e fornecendo uma avaliação mais realista do desempenho do sistema em condições operacionais reais;
- Inserção do histórico da conversa como uma forma de enriquecer os dados textuais, possibilitando deixar mais claro a diferença das categorias;
- Análise mais profunda das classes e suas características, possibilitando agrupar classes semelhantes.

REFERÊNCIAS

AHMED, H. *et al.* An investigation on disparity responds of machine learning algorithms to data normalization method. **ARO-THE SCIENTIFIC JOURNAL OF KOYA UNIVERSITY**, v. 10, p. 29–37, 09 2022.

ALEKSOVSKI, D.; KOCEV, D.; DŽEROSKI, S. Evaluation of distance measures for hierarchical multi-label classification in functional genomics. *In*: **Proceedings of the 1st workshop on learning from multi-label data (MLD) held in conjunction with ECML/PKDD**. [*S.l.: s.n.*], 2009. p. 5–16. Available at: <https://kt.ijs.si/dragi_kocev/old-web/bibliography/resources/2009MLD_HierarchicalDistances.pdf>.

ALMEIDA, T. *et al.* Text normalization and semantic indexing to enhance instant messaging and sms spam filtering. **Knowledge-Based Systems**, v. 108, 05 2016.

ALTUN, Y.; HOFMANN, T. Large margin methods for label sequence learning. *In*: **Interspeech**. [*S.l.: s.n.*], 2003. Available at: <https://api.semanticscholar.org/CorpusID:12006035>.

ALY, R.; REMUS, S.; BIEMANN, C. Hierarchical multi-label classification of text with capsule networks. *In*: **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop**. Florence, Italy: Association for Computational Linguistics, 2019. p. 323–330. Available at: <https://aclanthology.org/P19-2045>.

ANICK, P. G.; VAITHYANATHAN, S. Exploiting clustering and phrases for context-based information retrieval. **SIGIR Forum**, Association for Computing Machinery, New York, NY, USA, v. 31, n. SI, p. 314–323, jul. 1997. ISSN 0163-5840. Available at: <https://research.ibm.com/publications/exploiting-clustering-and-phrases-for-context-based-information-retrieval>.

APTÉ, C.; DAMERAU, F. J.; WEISS, S. M. Automated learning of decision rules for text categorization. **ACM Transactions on Information Systems**, 1994. Available at: <https://research.ibm.com/publications/automated-learning-of-decision-rules-for-text-categorization>.

BABBAR, R. *et al.* On flat versus hierarchical classification in large-scale taxonomies. *In*: **Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2**. Red Hook, NY, USA: Curran Associates Inc., 2013. (NIPS'13), p. 1824–1832.

BANERJEE, S. *et al.* Hierarchical transfer learning for multi-label text classification. *In*: KORHONEN, A.; TRAUM, D.; MÀRQUEZ, L. (ed.). **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**. Florence, Italy: Association for Computational Linguistics, 2019. p. 6295–6300.

BENGIO, Y. *et al.* Neural probabilistic language models. **J Mach Learn Res**, v. 3, p. 137–186, 05 2006. Available at: <https://www.jmlr.org/papers/volume3/bengio03a/bengio03a.pdf>.

BENNETT, P. N.; NGUYEN, N. Refined experts: improving classification in large taxonomies. *In*: **Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval**. New York, NY, USA: Association for Computing Machinery, 2009. (SIGIR '09), p. 11–18. ISBN 9781605584836. Available at: <https://doi.org/10.1145/1571941.1571946>.

BITTENCOURT, M. d. M. **ML-MDLText: um mètodo de classificaccão de textos multirròtulo de aprendizado incremental**. 2020. 138 p. Tese (Doutorado) — Universidade Federal de São Carlos – UFSCar, Sorocaba, SP, 2020. Available at: <https://repositorio.ufscar.br/handle/ufscar/12436>.

BLOCKEEL, H. *et al.* Hierarchical multi-classification. *In*: **Knowledge Discovery and Data Mining**. [*S.l.: s.n.*], 2002.

BLOCKEEL, H. *et al.* Decision trees for hierarchical multilabel classification: A case study in functional genomics. *In*: **Proceedings of 10th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2006)**. Berlin: Springer, 2006. (Lecture notes in Computer Science, v. 4213), p. 18–29. ISBN 978-3-540-45374-1.

BLUMBERG, R.; ATRE, S. The problem with unstructured data. *In*: . [*S.l.: s.n.*], 2003. p. 1. Available at: <https://atrepower.com/pdf/resources/DMReview/DMReviewFeb2003.pdf>.

BOJANOWSKI, P. *et al.* Enriching word vectors with subword information. **Transactions of the Association for Computational Linguistics**, MIT Press, Cambridge, MA, v. 5, p. 135–146, 2017. Available at: <https://aclanthology.org/Q17-1010>.

BREIMAN, L. Random forests. **Machine Learning**, Kluwer Academic Publishers, Dordrecht, Netherlands, v. 45, p. 5–32, 10 2001. Available at: <https://link.springer.com/article/10.1023/A:1010933404324>.

BREIMAN, L. *et al.* **Classification and Regression Trees**. 1. ed. [*S.l.: s.n.*]: Chapman and Hall/CRC, 1984. ISBN 9780412348200.

BROWN, T. B. *et al.* Language models are few-shot learners. *In*: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 33: ANNUAL CONFERENCE ON NEURAL INFORMATION PROCESSING SYSTEMS 2020. **Proceedings of the 34th International Conference on Neural Information Processing Systems**. Red Hook, NY, USA: Curran Associates Inc., 2020. (NIPS '20). ISBN 9781713829546. Available at: <https://arxiv.org/pdf/2005.14165>.

CAI, L.; HOFMANN, T. Hierarchical document categorization with support vector machines. *In*: **International Conference on Information and Knowledge Management**. New York, NY, USA: Association for Computing Machinery, 2004. (CIKM '04), p. 78–87. ISBN 1581138741. Available at: <https://doi.org/10.1145/1031171.1031186>.

CASELI, H. M.; NUNES, M. G. V. **Processamento de Linguagem Natural: Conceitos, Técnicas e Aplicações em Português**. 2. ed. São Carlos: Biblioteca da Sociedade Brasileira de Processamento de Linguagem Natural (BPLN), 2023. ISBN 978-65-00-95750-1. Available at: <https://brasileiraspln.com/livro-pln/2a-edicao>.

CECI, M.; MALERBA, D. Classifying web documents in a hierarchy of categories: A comprehensive study. **J. Intell. Inf. Syst.**, v. 28, p. 37–78, 02 2007.

CERRI, R.; BARROS, R. C.; CARVALHO, A. C. P. L. F. de. Hierarchical classification of gene ontology-based protein functions with neural networks. *In*: **2015 International Joint Conference on Neural Networks (IJCNN)**. [*S.l.: s.n.*], 2015. p. 1–8.

CESA-BIANCHI, N.; GENTILE, C.; ZANIBONI, L. Hierarchical classification: combining bayes with svm. *In*: **Proceedings of the 23rd International Conference on Machine Learning**. New York, NY, USA: Association for Computing Machinery, 2006. (ICML '06), p. 177–184. ISBN 1595933832. Available at: <https://doi.org/10.1145/1143844.1143867>.

CESA-BIANCHI, N.; GENTILE, C.; ZANIBONI, L. Incremental algorithms for hierarchical classification. **Journal of Machine Learning Research**, v. 7, p. 31–54, 12 2006.

CHAKRABARTI, S. *et al.* Scalable feature selection, classification and signature generation for organizing large text databases into hierarchical topic taxonomies. **The VLDB Journal**, Springer-Verlag, Berlin, Heidelberg, v. 7, p. 163–178, 10 1998. ISSN 1066-8888. Available at: <https://www.researchgate.net/publication/2457587_Scalable_feature_selection_classification_and_signature_generation_for_organizing_large_text_databases_into_hierarchical_topic_taxonomies>.

CHEN, B. *et al.* Hyperbolic interaction model for hierarchical multi-label classification. **Proceedings of the AAAI Conference on Artificial Intelligence**, Association for the Advancement of Artificial Intelligence (AAAI), v. 34, n. 05, p. 7496–7503, abr. 2020. ISSN 2159-5399. Available at: <http://dx.doi.org/10.1609/AAAI.V34I05.6247>.

CHEN, H. *et al.* Hierarchy-aware label semantics matching network for hierarchical text classification. *In*: **Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing**. Bangkok, Thailand: Association for Computational Linguistics, 2021. p. 4370–4379.

CHOMSKY, N. **Syntactic Structures**. 2. ed. Berlim: Walter de Gruyter, 1957/2002. ISBN 3-11-017279-8. Available at: <https://www.ling.upenn.edu/courses/ling5700/Chomsky1957.pdf>.

CHURCH, K. W.; HANKS, P. Word association norms, mutual information, and lexicography. **Computational Linguistics**, MIT Press, Cambridge, MA, USA, v. 16, n. 1, p. 22–29, mar. 1990. ISSN 0891-2017. Available at: <https://www.researchgate.net/publication/2477223_Word_Association_Norms_Mutual_Information_and_Lexicography>.

CLARE, A.; KING, R. Predicting gene function in saccharomyces cerevisiae. **Bioinformatics (Oxford University Press)**, Oxford, England, v. 19, n. 2, p. 42–49, 11 2003.

COLLOBERT, R.; WESTON, J. A unified architecture for natural language processing: deep neural networks with multitask learning. *In*: **Proceedings of the 25th International Conference on Machine Learning**. New York, NY, USA: Association for Computing Machinery, 2008. (ICML '08), p. 160–167. ISBN 9781605582054. Available at: <https://doi.org/10.1145/1390156.1390177>.

CORTES, C.; VAPNIK, V. N. Support-vector networks. **Machine Learning**, Kluwer Academic Publishers, USA, v. 20, n. 3, p. 273–297, 1995. ISSN 0885-6125. Available at: <https://link.springer.com/article/10.1007/BF00994018>.

COSTA, E. P. *et al.* Comparing several approaches for hierarchical classification of proteins with decision trees. *In*: SAGOT, M.-F.; WALTER, M. E. M. T. (ed.). **Advances in Bioinformatics and Computational Biology**. Berlin, Heidelberg: Springer, 2007, (Lecture Notes in Computer Science, v. 4643). Available at: <(linkunavailable)>.

COVER, T. M.; HART, P. E. Nearest neighbor pattern classification. **IEEE Transaction on Information Theory**, IEEE Press, v. 13, n. 1, p. 21–27, set. 2006. ISSN 0018-9448. Available at: <https://link.springer.com/chapter/10.1007/978-3-540-74976-9_25>.

D'ALESSIO, S. *et al.* The effect of using hierarchical classifiers in text categorization. *In*: **Content-Based Multimedia Information Access - Volume 1**. Paris, FRA: LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE, 2000. (RIAO '00), p. 302–313. Available at: <https://dl.acm.org/doi/pdf/10.5555/2835865.2835898>.

DAVIS, S. P. Hilcc: A hierarchical interface to library of congress classification. **Journal of Internet Cataloging**, v. 5, n. 4, p. 19–49, 2002. Available at: <https://academiccommons.columbia.edu/doi/10.7916/D8RV0ZNK/download>.

DEKEL, O.; KESHET, J.; SINGER, Y. Large margin hierarchical classification. **21st International Conference on Machine Learning (ICML '04)**, Association for Computing Machinery (ACM), Banff, Alberta, Canadá, p. 27, 09 2004.

DENG, Z. *et al.* Htcinfomax: A global model for hierarchical text classification via information maximization. *In*: **Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**. Online: Association for Computational Linguistics, 2021. p. 3259–3265. Available at: <http://dx.doi.org/10.18653/V1/2021.NAACL-MAIN.260>.

DEVLIN, J. *et al.* BERT: Pre-training of deep bidirectional transformers for language understanding. *In*: BURSTEIN, J.; DORAN, C.; SOLORIO, T. (ed.). **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)**. Minneapolis, Minnesota: Association for Computational Linguistics, 2019. p. 4171–4186. Available at: <https://aclanthology.org/N19-1423>.

DIETTERICH, T. G. Machine learning research: Four current directions. **AI Magazine**, v. 18, n. 4, p. 97–136, 1997. Available at: <https://onlinelibrary.wiley.com/doi/full/10.1609/aimag.v18i4.1324>.

DUMAIS, S. T.; CHEN, H. Hierarchical classification of web content. *In*: **Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval**. New York, NY, USA: Association for Computing Machinery, 2000. (SIGIR '00), p. 256–263. ISBN 1581132263. Available at: <https://doi.org/10.1145/345508.345593>.

EISNER, R. *et al.* Improving protein function prediction using the hierarchical structure of the gene ontology. *In*: **2005 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology**. [*S.l.: s.n.*], 2005. p. 1–10.

FACELI, K. L. *et al.* **Inteligência artificial: uma abordagem de aprendizado de máquina**. Rio de Janeiro: LTC, 2011. 394 p. ISBN 9788521618805.

FAGNI, T.; SEBASTIANI, F. On the selection of negative examples for hierarchical text categorization. **Proceedings 3rd Lang Technology Conference**, p. 24–28, 01 2007. Available at: <chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/http://nmis.isti.cnr.it/sebastiani/Publications/LTC07b.pdf>.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. The kdd process for extracting useful knowledge from volumes of data. **Commun. ACM**, Association for Computing Machinery, New York, NY, USA, v. 39, n. 11, p. 27–34, nov. 1996. ISSN 0001-0782. Available at: <https://doi.org/10.1145/240455.240464>.

FRAKES, W. B.; BAEZA-YATES, R. (ed.). **Information retrieval: data structures and algorithms**. USA: Prentice-Hall, Inc., 1992. ISBN 0134638379.

FREITAS, A. A.; CARVALHO, A. C. P. L. F. A tutorial on hierarchical classification with applications in bioinformatics. *In*: TANIAR, D. (ed.). **Research and Trends in Data Mining Technologies and Applications**. [*S.l.: s.n.*]: IGI Global, 2007. p. 175–208. ISBN 9781599042718.

FREUND, Y.; SCHAPIRE, R. E. Experiments with a new boosting algorithm. *In*: **Proceedings of the Thirteenth International Conference on International Conference on Machine Learning**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1996. (ICML'96), p. 148–156. ISBN 1558604197.

GALVAO, L. R.; MERSCHMANN, L. H. C. Hsim: A supervised imputation method for hierarchical classification scenario. *In*: **Discovery Science: 19th International Conference, DS 2016, Bari, Italy, October 19–21, 2016, Proceedings**. Berlin, Heidelberg: Springer-Verlag, 2016. p. 134–148. ISBN 978-3-319-46306-3. Available at: <https://doi.org/10.1007/978-3-319-46307-0_9>.

GANTZ, J.; REINSEL, D. Relatório técnico, **The Digital Universe in 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East**. 2012. Available at: <https://www.cs.princeton.edu/courses/archive/spring13/cos598C/idc-the-digital-universe-in-2020.pdf>.

GÉRON, A. **Mãos à obra Aprendizado de Máquina com Skcikit-Learn & TensorFlow: Conceitos, Ferramentas e Técnicas para a Construção de Sistemas Inteligentes**. [*S.l.: s.n.*]: Alta Books, 2019. ISBN 9788550803814.

GOPAL, S.; YANG, Y. Recursive regularization for large-scale classification with hierarchical and graphical dependencies. *In*: **Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. New York, NY, USA: Association for Computing Machinery, 2013. (KDD '13), p. 257–265. ISBN 9781450321747. Available at: <https://doi.org/10.1145/2487575.2487644>.

HAPKE, H. M.; HOWARD, C.; LANE, H. Understanding, analyzing, and generating text with python. *In*: **Natural Language Processing in Action**. [*S.l.: s.n.*]: Manning Publications, 2019. ISBN 9781617294631.

HAYKIN, S. **Neural Networks: A Comprehensive Foundation**. 2. ed. USA: Prentice Hall PTR, 1998. ISBN 0132733501.

HOLDEN, N.; FREITAS, A. A. Improving the performance of hierarchical classification with swarm intelligence. *In*: **Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics (EvoBio)**. Berlin: Springer, 2008. (Lecture Notes in Computer Science, v. 4973), p. 48–60.

IRSAN, I. C.; KHODRA, M. L. Hierarchical multilabel classification for indonesian news articles. *In*: **2016 International Conference on Advanced Informatics: Concepts, Theory and Application (ICAICTA)**. [*S.l.: s.n.*], 2016.

KOLLER, D.; SAHAMI, M. Hierarchically classifying documents using very few words. *In*: **International Conference on Machine Learning**. [*S.l.: s.n.*], 1997. Available at: <https://api.semanticscholar.org/CorpusID:2112467>.

KRENDZELAK, M.; JAKAB, F. Hierarchical text classification using cnns with local classification per parent node approach. *In*: **2019 17th International Conference on Emerging eLearning Technologies and Applications (ICETA)**. [*S.l.: s.n.*], 2019. p. 460–464.

LARKEY, L. S. Some issues in the automatic classification of u.s. patents. *In*: **Learning for Text Categorization**. Amherst, MA: AAAI, 1998, (AAAI Technical Report WS-98-05). Available at: <https://cdn.aaai.org/Workshops/1998/WS-98-05/WS98-05-015.pdf>.

LIMA, H. C. S. C. *et al.* A vns algorithm for feature selection in hierarchical classification context. **Electronic Notes in Discrete Mathematics**, v. 66, p. 79–86, 04 2018.

LIMA, H. C. S. C. *et al.* A novel hybrid feature selection algorithm for hierarchical classification. **IEEE Access**, v. 9, p. 127278–127292, 2021. Available at: <https://ieeexplore.ieee.org/document/9536739>.

LING, W. *et al.* Two/too simple adaptations of word2vec for syntax problems. *In*: MIHALCEA, R.; CHAI, J.; SARKAR, A. (ed.). **Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**. Denver, Colorado: Association for Computational Linguistics, 2015. p. 1299–1304. Available at: <https://aclanthology.org/N15-1142>.

LOVINS, J. B. Development of a stemming algorithm. **Mechanical Translation and Computational Linguistics**, v. 11, p. 22–31, 1968. Available at: <https://aclanthology.org/www.mt-archive.info/MT-1968-Lovins.pdf>.

MANNING, C. D.; RAGHAVAN, P.; SCHÜTZE, H. **Introduction to Information Retrieval**. Cambridge University Press, 2009. Available at: <https://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf>.

MAO, Y. *et al.* Hierarchical text classification with reinforced label assignment. *In*: **Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)**. Hong Kong, China: Association for Computational Linguistics, 2019. p. 445–455. Available at: <http://dx.doi.org/10.18653/v1/D19-1042>.

MCCALLUM, A.; NIGAM, K. A comparison of event models for naive bayes text classification. *In*: **Proceedings of the AAAI-98 Workshop on Learning for Text Categorization**. Madison: [*S.l.: s.n.*], 1998. p. 41–48. Available at: <https://cdn.aaai.org/Workshops/1998/WS-98-05/WS98-05-007.pdf>.

MCCALLUM, A. *et al.* Improving text classification by shrinkage in a hierarchy of classes. *In*: **International Conference on Machine Learning**. [*S.l.: s.n.*], 1998. Available at: <https://api.semanticscholar.org/CorpusID:9086884>.

MENG, Y. *et al.* Weakly-supervised hierarchical text classification. **Proceedings of the AAAI Conference on Artificial Intelligence**, Association for the Advancement of Artificial Intelligence (AAAI), v. 33, n. 01, p. 6826–6833, jul. 2019. ISSN 2159-5399. Available at: <https://dl.acm.org/doi/pdf/10.1609/aaai.v33i01.33016826>.

METZ, J. **Abordagens para aprendizado semissupervisionado multirrótulo e hierárquico**. 2011. Tese (Doutorado) — Universidade de São Paulo, São Carlos, 2011. Available at: <http://www.teses.usp.br/teses/disponiveis/55/55134/tde-13012012-144607/>.

MIKOLOV, T. *et al.* Distributed representations of words and phrases and their compositionality. *In*: **Proceedings of the 26th International Conference on Neural Information Processing Systems**. Red Hook, NY, USA: Curran Associates Inc., 2013. (NIPS'13, v. 2), p. 3111–3119. Available at: <https://proceedings.neurips.cc/paper_files/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf>.

MIRANDA, F. M.; KÖHNECKE, N.; RENARD, B. Y. Hiclass: A python library for local hierarchical classification compatible with scikit-learn. **Journal of Machine Learning Research**, v. 24, p. 1–17, jan. 2023. Editor: Alexandre Gramfort.

MITCHELL, T. M. **Machine Learning**. 1. ed. USA: McGraw-Hill, Inc., 1997. ISBN 0070428077.

MNIH, A.; HINTON, G. A scalable hierarchical distributed language model. *In*: . [*S.l.: s.n.*], 2008. p. 1081–1088.

NAIK, A.; RANGWALA, H. Large scale hierarchical classification: State of the art. *In*: SPRINGERBRIEFS IN COMPUTER SCIENCE. [*S.l.: s.n.*], 2018. p. 15,18,19,20,21,22,23.

NIGAM, K. P. **Using unlabeled data to improve text classification**. 2001. 138 p. Tese (Doutorado) — Carnegie Mellon University, Pittsburgh, PA, 2001. Available at: <https://www.proquest.com/openview/282d9f7565185cd07cf9c8296a8e824f/1?pq-origsite=gscholar&cbl=18750&diss=y>.

PAES, B. C.; PLASTINO, A.; FREITAS, A. A. Improving local per level hierarchical classification. **Journal of Information and Data Management**, v. 3, n. 3, p. 394, 2012. Available at: <(linkunavailable)>.

PARTALAS, I. *et al.* Lshtc: A benchmark for large-scale text classification. *In*: . [*S.l.: s.n.*], 2015. Available at: <https://api.semanticscholar.org/CorpusID:17350068>.

PENNINGTON, J.; SOCHER, R.; MANNING, C. Glove: Global vectors for word representation. *In*: . [*S.l.: s.n.*], 2014. v. 14, p. 1532–1543.

PERDIH, T. S. *et al.* Option predictive clustering trees for hierarchical multi-label classification. *In*: **International Conference on Discovery Science**. [*S.l.: s.n.*], 2017. (Lecture notes in Computer Science), p. 116–123.

PEREIRA, R. M.; COSTA, Y. M. G.; JR., C. N. S. Handling imbalance in hierarchical classification problems using local classifiers approaches. **Data Mining and Knowledge Discovery**, Springer Science+Business Media LLC, v. 35, n. 4, p. 1221–1245, 2021.

PETERS, M. E. *et al.* Deep contextualized word representations. *In*: . [*S.l.: s.n.*], 2018.

QIU, X.; GAO, W.; HUANG, X. Hierarchical multi-label text categorization with global margin maximization. *In*: **Proceedings of the ACL-IJCNLP 2009 Conference Short Papers**. Stroudsburg, PA, EUA: Association for Computational Linguistics, 2009. p. 165–168.

RAMÍREZ-CORONA, M.; SUCAR, L.; MORALES, E. Hierarchical multilabel classification based on path evaluation. **International Journal of Approximate Reasoning**, v. 68, 07 2016.

READ, J. *et al.* Sentence boundary detection: A long solved problem? *In*: KAY, M.; BOITET, C. (ed.). **Proceedings of COLING 2012: Posters**. Mumbai, India: The COLING 2012 Organizing Committee, 2012. p. 985–994. Available at: <https://aclanthology.org/C12-2096>.

REINSEL, D.; GANTZ, J.; RYDNING, J. **The Digitization of the World: From Edge to Core**. [*S.l.*], 2018. Available at: <https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf>.

RENNIE, J. D. M. *et al.* Tackling the poor assumptions of naive bayes text classifiers. *In*: . [*S.l.: s.n.*], 2003. v. 41, p. 616–623.

RODRÍGUEZ-CANTELAR, M. *et al.* Automatic detection of inconsistencies and hierarchical topic classification for open-domain chatbots. *In*: . [*S.l.: s.n.*], 2023.

ROUSU, J. *et al.* Learning hierarchical multi-category text classification models. **Proceedings of the 2nd International Conference on Machine Learning**, Bonn, Alemanha, p. 744–751, 08 2005.

ROUSU, J. *et al.* Kernel-based learning of hierarchical multilabel classification models. **Journal of Machine Learning Research**, JMLR.org, v. 7, p. 1601–1626, 12 2006. ISSN 1532-4435. Available at: <https://www.researchgate.net/publication/221996780_Kernel-Based_Learning_of_Hierarchical_Multilabel_Classication_Models>.

RUIZ, M. E.; SRINIVASAN, P. Hierarchical text categorization using neural networks. **Information Retrieval**, v. 5, p. 87–118, 01 2002. Available at: <https://link.springer.com/article/10.1023/A:1012782908347>.

SAMUEL, A. L. Some studies in machine learning using the game of checkers. *In*: . [*S.l.: s.n.*], 1959. v. 3, n. 3, p. 206–226.

SANTOS, B. Z. *et al.* Predictive bi-clustering trees for hierarchical multi-label classification. *In*: **Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part III**. Berlin, Heidelberg: Springer-Verlag, 2020. p. 701–718. ISBN 978-3-030-67663-6. Available at: <https://doi.org/10.1007/978-3-030-67664-3_42>.

SEBASTIANI, F. Machine learningin automated text categorization. *In*: ACM COMPUTING SURVEYS. [*S.l.: s.n.*], 2001. p. 1,2.

SECKER, A. *et al.* An experimental comparison of classification algorithms for the hierarchical prediction of protein function. **Expert Update (the BCS-SGAI Magazine)**, v. 9, p. 17–22, 01 2007.

SEEGER, M. W. Cross-validation optimization for large scale structured classification kernel methods. **Journal of Machine Learning Research**, v. 9, p. 1147–1178, 06 2008.

SILLA, C. N.; FREITAS, A. A. A global-model naive bayes approach to the hierarchical prediction of protein functions. *In*: **2009 Ninth IEEE International Conference on Data Mining**. [*S.l.: s.n.*], 2009. p. 549–554. Available at: <https://www.cs.kent.ac.uk/people/staff/aaf/pub_papers.dir/IEEE-ICDM-2009-Silla.pdf>.

SILLA, C. N.; FREITAS, A. A. A survey of hierarchical classification across different application domains. **Data Mining and Knowledge Discovery**, Springer, v. 22, n. 1-2, p. 31–72, 2011.

SILVA, C. O.; SUGUIY, R. T.; SILLA, C. N. An investigation of different positive and negative training policies for the task of hierarchical music genre classification. *In*: **2023 30th International Conference on Systems, Signals and Image Processing (IWSSIP)**. [*S.l.: s.n.*], 2023. p. 1–5.

SMOLA, A. J. *et al.* Advances in large margin classifiers. *In*: MIT PRESS. [*S.l.: s.n.*], 2000.

SRI, M. Practical natural language processing with python. *In*: APRESS. [*S.l.: s.n.*], 2020.

SUJON, K. M. *et al.* When to use standardization and normalization: Empirical evidence from machine learning models and xai. **IEEE Access**, v. 12, p. 135300–135314, 2024.

SUN, A.; LIM, E.-P. Hierarchical text classification and evaluation. *In*: **Proceedings 2001 IEEE International Conference on Data Mining (ICDM '01)**. Washington, DC, EUA: IEEE Computer Society, 2001. p. 521–528. ISBN 0769511198.

TABASSUM, A.; PATIL, R. R. A survey on text pre-processing & feature extraction techniques in natural language processing. *In*: . [*S.l.: s.n.*], 2020. v. 7. Available at: <https://www.academia.edu/download/64643112/IRJET-V7I6913.pdf>.

TAVARES, L. L. **Utilização de mecanismos de roteamento para seleção de sistemas de Question Answering**. 2018. candthesis — Universidade Federal de São Carlos Câmpus Sorocaba, 2018. Available at: <https://repositorio.ufscar.br/handle/ufscar/10278>.

TIKK, D.; BIRÓ, G.; YANG, J. A hierarchical text categorization approach and its application to frt expansion. **Australian Journal of Intelligent Information Processing Systems**, v. 8, n. 3, p. 123–131, 04 2003. Available at: <https://users.softlab.ntua.gr/facilities/public/AD/Text%20Categorization/A%20hierarchical%20text%20categorization%20approach%20and%20its%20application%20to%20FRT%20expansion.pdf>.

TRASK, A.; MICHALAK, P.; LIU, J. sense2vec - a fast and accurate method for word sense disambiguation in neural word embeddings. *In*: . [*S.l.: s.n.*], 2015.

TSOCHANTARIDIS, I. *et al.* Large margin methods for structured and interdependent output variables. **J. Mach. Learn. Res.**, v. 6, p. 1453–1484, 2005. Available at: <https://www.jmlr.org/papers/volume6/tsochantaridis05a/tsochantaridis05a.pdf>.

TURIAN, J.; RATINOV, L.-A.; BENGIO, Y. Word representations: A simple and general method for semi-supervised learning. *In*: HAJIČ, J. *et al.* (ed.). **Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics**. Uppsala, Sweden: Association for Computational Linguistics, 2010. p. 384–394. Available at: <https://aclanthology.org/P10-1040>.

UYSAL, A. K.; GUNAL, S. The impact of preprocessing on text classification. *In*: . [*S.l.: s.n.*], 2014. v. 50, p. 104–112.

VAITHYANATHAN, S.; MAO, J.; DOM, B. Hierarchical bayes for text classification. *In*: . [*S.l.: s.n.*], 2000.

VAPNIK, V. N. An overview of statistical learning theory. **IEEE transactions on neural networks**, v. 10, n. 5, p. 988–999, 1999. Available at: <https://api.semanticscholar.org/CorpusID:6294728http://www.mit.edu/~6.454/www_spring_2001/emin/slt.pdf>.

VASWANI, A. *et al.* All mistakes are not equal: Comprehensive hierarchy aware multi-label predictions (champ). **ArXiv**, abs/2206.08653, 2022.

VENS, C. *et al.* Decision trees for hierarchical multi-label classification. **Machine Learning**, v. 73, p. 185–214, 11 2008.

WANG, L. *et al.* Multilingual e5 text embeddings: A technical report. **arXiv preprint arXiv:2402.05672**, 2024.

WANG, Y.; HOU, Y.; CHE WANXIANGAND LIU, T. From static to dynamic word representations: a survey. *In*: . [*S.l.: s.n.*]: International Journal of Machine Learning and Cybernetics, 2020.

WANG, Z. *et al.* Incorporating hierarchy into text encoder: a contrastive learning approach for hierarchical text classification. *In*: **Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**. Dublin, Ireland: Association for Computational Linguistics, 2022. p. 7109–7119. Available at: <https://aclanthology.org/2022.acl-long.491/>.

WANG, Z. *et al.* Hpt: Hierarchy-aware prompt tuning for hierarchical text classification. *In*: **In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing**. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, 2022. p. 3740–3751.

WEISS, S. M.; INDURKHYA, N.; ZHANG, T. Fundamentals of predictive text mining. *In*: **Texts in Computer Science**. [*S.l.: s.n.*], 2010. Available at: <https://api.semanticscholar.org/CorpusID:2674061>.

WIENER, E. D.; PEDERSEN, J. O.; WEIGEND, A. S. A neural network approach to topic spotting. *In*: **4th Annual Symposium on Document Analysis and Information Retrieval**. [*S.l.: s.n.*], 1995. (SDAIR-95, v. 317), p. 332. Available at: <chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=abbe40b7503f51971c92f9f9b20ebea6c0b36d77>.

WILBUR, W. J.; KIM, W. The ineffectiveness of within-document term frequency in text classification. *In*: INFORMATION RETRIEVAL. 2009. v. 12, n. 5, p. 509–525. Available at: <https://www.researchgate.net/publication/26868618_The_Ineffectiveness_of_Within_-_Document_Term_Frequency_in_Text_Classification>.

WU, F.; ZHANG, J.; HONAVAR, V. Learning classifiers using hierarchically structured class taxonomies. *In*: ZUCKER, J.; SAITTA, L. (ed.). **Abstraction, Reformulation and Approximation**. Berlin, Heidelberg: Springer, 2005. (Lecture Notes in Computer Science, v. 3607), p. 313–320.

XIAO, Z. *et al.* Automatic hierarchical classification of emotional speech. *In*: **Ninth IEEE International Symposium on Multimedia Workshops (ISMW 2007)**. IEEE, 2007. v1, p. 291–296. Available at: <https://liris.cnrs.fr/Documents/Liris-2742.pdf>.

XU, Z. *et al.* An alternative text representation to tf-idf and bag-of-words. *In*: . [*S.l.: s.n.*], 2013. Available at: <https://arxiv.org/abs/1301.6770>.

ZHAN, W.; YOSHIDA, T.; TANG, X. A comparative study of tfidf, lsi and multi-words for text classification. *In*: EXPERT SYSTEMS WITH APPLICATIONS. [*S.l.: s.n.*], 2011. v. 38, n. 3, p. 2758–2765.

ZHANG, X. *et al.* La-hcn: Label-based attention for hierarchical multi-label text classification neural network. **Expert Systems with Applications**, Elsevier BV, v. 187, p. 115922, jan. 2022. ISSN 0957-4174. Available at: <http://dx.doi.org/10.1016/j.eswa.2021.115922>.

ZHANG, Y. *et al.* Match: Metadata-aware text classification in a large hierarchy. *In*: **Proceedings of the Web Conference 2021**. ACM, 2021. (WWW '21), p. 3246–3257. Available at: <http://dx.doi.org/10.1145/3442381.3449979>.

ZHAO, R. *et al.* Hierarchical multi-label text classification: Self-adaption semantic awareness network integrating text topic and label level information. *In*: **Knowledge Science, Engineering and Management: 14th International Conference, KSEM 2021, Tokyo, Japan, August 14–16, 2021, Proceedings, Part II**. Berlin, Heidelberg: Springer-Verlag, 2021. p. 406–418. ISBN 978-3-030-82146-3. Available at: <https://doi.org/10.1007/978-3-030-82147-0_33>.

ZHENG, W.; ZHAO, H. Cost-sensitive hierarchical classification for imbalance classes. **Applied Intelligence**, Kluwer Academic Publishers, USA, v. 50, n. 8, p. 2328–2338, ago. 2020. ISSN 0924-669X. Available at: <https://doi.org/10.1007/s10489-019-01624-z>.

ZHOU, J. *et al.* Hierarchy-aware global model for hierarchical text classification. *In*: **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**. Online: Association for Computational Linguistics, 2020. p. 1106–1117. Available at: <https://aclanthology.org/2020.acl-main.104>.

ZHU, H. *et al.* Hill: Hierarchy-aware information lossless contrastive learning for hierarchical text classification. *In*: **In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)**. Mexico City, Mexico: Association for Computational Linguistics, 2024. p. 4731–4745.

ZHU, H. *et al.* Hitin: Hierarchy-aware tree isomorphism network for hierarchical text classification. **ArXiv**, abs/2305.15182, p. 7809–7821, 05 2023. Available at: <https://doi.org/10.48550/arXiv.2305.15182>.