VÂNIA BATISTA DOS SANTOS

META-APRENDIZAGEM APLICADA A CLASSIFICAÇÃO DE TEXTO MULTIRRÓTULO

LAVRAS – MG

2020

VÂNIA BATISTA DOS SANTOS

META-APRENDIZAGEM APLICADA A CLASSIFICAÇÃO DE TEXTO MULTIRRÓTULO

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, área de concentração em Inteligência Computacional e Processamento Gráfico, para obtenção do título de Mestre.

Prof. Dr. Luiz Henrique de Campos Merschmann
Orientador

LAVRAS – MG

2020

Ficha catalográfica elaborada pela Coordenadoria de Processos Técnicos da Biblioteca Universitária da UFLA

Santos, Vânia Batista dos.

Meta-aprendizagem Aplicada a Classificação de Texto Multirrótulo / Vânia Batista dos Santos. – Lavras : UFLA, 2020.

73 p. : il.

Dissertação (mestrado acadêmico)–Universidade Federal de Lavras, 2020.

Orientador: Prof. Dr. Luiz Henrique de Campos Merschmann .

Bibliografia.

1. Meta-aprendizagem. 2. Classificação de Texto. 3. Classificação Multirrótulo. I. Merschmann, Luiz Henrique de Campos. II. Título.

VÂNIA BATISTA DOS SANTOS

META-APRENDIZAGEM APLICADA A CLASSIFICAÇÃO DE TEXTO MULTIRRÓTULO
METALEARNING APPLIED TO MULTI-LABEL TEXT CLASSIFICATION

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, área de concentração em Inteligência Computacional e Processamento Gráfico, para obtenção do título de Mestre.

APROVADA em 21 de Fevereiro de 2020.

Prof. Dr. Denilson Alves Pereira	UFLA
Profa. Dra. Elaine Ribeiro de Faria Paiva	UFU

Prof. Dr. Luiz Henrique de Campos Merschmann
Orientador

LAVRAS – MG
2020

Dedico esta dissertação à Deus, minha fortaleza! À minha família, em especial aos meus pais, Sebastião e Maria (in memoriam), fontes de inspiração e força, cujos amor e dedicação moldaram minha trajetória.

AGRADECIMENTOS

Grata sou, impreterivelmente, à Deus pela oportunidade de concluir mais uma etapa em minha formação educacional.

Agradeço aos meus pais, Sebastião e Maria (*in memoriam*), meus primeiros e maiores mestres. Em especial, agradeço minha mãe, por ter sonhado com uma educação de qualidade para mim e, quando em vida, não ter medido esforços para garantir que eu tivesse condições para obtê-la. Agradeço aos meus irmãos pela torcida incessante e por existirem em minha vida! Em especial, agradeço a minha irmã Cátia, por sempre acreditar no meu potencial e ser apoio e incentivo nos momentos de decisão.

Um agradecimento especial à todos os professores que passaram por minha vida e compartilharam seus saberes, principalmente aqueles que deixaram ensinamentos para a vida. Em especial, agradeço as Professoras Herlita Mourão e Karina Dutra.

Agradeço também aos meus amigos do mestrado: Douglas Nunes, por ser sempre solícito e compartilhar conhecimento; Elena e Juliana, pela força, amizade e carinho de sempre; Gustavo, Daiane, Lucas, Bia, Vitor, Natana, Leonardo, Fernando, Luana, Douglas Silva, Italo, César, Thauane e a todos aqueles que conseguiram tornar a caminhada mais valiosa. Obrigada!

Agradeço ao meu orientador, Professor Luiz Henrique de Campos Merschmann, pela paciência, dedicação, comprometimento com nosso trabalho e por ter acreditado em mim. Agradeço também aos professores Denilson Pereira e Eric Araújo, que compuseram minha banca de qualificação e contribuíram para o desenvolvimento desta dissertação.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001 e da Universidade Federal de Lavras (UFLA). Portanto, agradeço pela oportunidade de realizar o mestrado nessa instituição e pelo apoio financeiro de ambas instituições. Assim como agradeço ao Departamento de Ciência da Computação (DCC) e a todos os seus funcionários, especialmente a secretária e amiga Luiza Junqueira.

Por fim, agradeço a todos que direta ou indiretamente contribuiram para a conclusão desta etapa, bem como aos que sempre torceram e torcem por mim. Muito obrigada!

"Prepara-se o cavalo para o dia da batalha, mas é do Senhor que depende a vitória."

(Provérbios, 21 - Bíblia Ave Maria)

RESUMO

A classificação é uma tarefa preditiva da Mineração de Dados que utiliza um conjunto de dados (instâncias), previamente rotulados, para o treinamento de um algoritmo que tem a função de aprender com os dados apresentados e ser capaz de predizer os rótulos de novas instâncias. A classificação multirrótulo, por sua vez, difere da tradicional classificação monorrótulo ao permitir que cada instância do conjunto de dados esteja associada a mais de um rótulo. Dessa forma, no domínio de textos, caracterizado pela diversidade, volume e produção crescente, a classificação multirrótulo desempenha um papel importante ao permitir que, de forma automática, seja extraído o máximo de informação embutida nesses dados. Por meio dos textos, é possível identificar conteúdos de grande valor para a tomada de decisão, como interesses, opiniões e sentimentos. Algumas áreas que exploram formas de trabalhar com os dados textuais são Processamento de Linguagem Natural, Mineração de Dados e Aprendizado de Máquina. A classificação multirrótulo dispõe de um significativo número de técnicas de aprendizagem disponíveis para a sua execução. Porém, encontrar a que seja mais apropriada para um determinado conjunto de dados, não é uma tarefa trivial, pois exige conhecimento das técnicas, consecutivos experimentos e, consequentemente, tempo. Nesse contexto, a meta-aprendizagem apresenta a relevância de sua aplicação ao investigar formas de automatizar o processo de seleção das melhores técnicas para determinado problema. Portanto, o objetivo deste trabalho é aplicar a meta-aprendizagem no desenvolvimento de um método para a classificação de textos multirrótulo, o qual busca selecionar o melhor algoritmo de classificação para cada instância do conjunto de dados apresentado. Os resultados experimentais demonstraram a eficácia do método proposto.

Palavras-chave: Meta-aprendizagem. Classificação de Texto. Classificação Multirrótulo. Processamento de Linguagem Natural. Mineração de dados.

ABSTRACT

Classification is a predictive task of Data Mining that uses a set of data (instances), previously labeled, to train an algorithm that has the function of learning from the data presented and being able to predict the labels of new instances. The multi-label classification, in turn, differs from the traditional single-label classification in that it allows each instance of the dataset to be associated with more than one label. Thus, in the domain of texts, characterized by diversity, volume, and increasing production, the multi-label classification plays an important role in allowing the maximum amount of information embedded in this data to be automatically extracted. Through texts, it is possible to identify the contents of great value for decision making, such as interests, opinions, and feelings. Some areas that explore ways to work with textual data are Natural Language Processing, Data Mining, and Machine Learning. The multi-label classification has a significant number of learning techniques available for its execution. However, finding the one that is most appropriate for a given dataset is not a trivial task, as it requires knowledge of techniques, consecutive experiments and, consequently, time. In this context, metalearning shows the relevance of its application when investigating ways to automate the process of selecting the best techniques for a given problem. Therefore, the objective of this work is to apply metalearning in the development of a method for the classification of multi-label texts, which seeks to select the best classification algorithm for each instance of the presented dataset. The experimental results demonstrated the effectiveness of the proposed method.

Keywords: Metalearning. Text Classification. Multi-label Classification. Natural Language Processing. Data Mining.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

A classificação de textos desempenha um papel muito importante na mais recente revolução tecnológica conhecida como Era da Informação. A tarefa de classificar textos, inicialmente aplicada principalmente em sistemas bibliotecários, permite a organização e a agilidade de acesso a informação e facilita a comunicação. Com a criação da *World Wide Web* (WWW), inventada em 1989 e disponibilizada em 1991, houve uma crescente produção dos mais diversos tipos de dados, que em grande parte são dados textuais não estruturados. Desta forma, diante desse grande repositório de dados, são necessárias técnicas automatizadas que possam extrair conhecimento a partir dos mesmos (BAEZA-YATES; RIBEIRO-NETO, 2011).

A popularização dos computadores, aliada às redes sociais e ao mercado digital, tem contribuído para que usuários se comuniquem e realizem suas compras pela Internet. Assim, muitas vezes por meio de textos, eles expressam seus sentimentos, interesses e opiniões sobre os mais variados assuntos. Por meio da integração de vários usuários nas redes sociais e a comunicação entre eles, o processo de promoção de *marketing* é facilitado. Com isso, identificar as tendências e comportamentos dos usuários é crucial para lidar com a concorrência e melhor atender as necessidades das partes interessadas. Essa realidade é o que desperta a atenção dos empresários, que visam conhecer os interesses e *feedbacks* de seus clientes (GALLINUCCI et al., 2015; MORO et al., 2016; AMADO et al., 2017). Além disso, a crescente produção de dados textuais está presente em diversos domínios, como acadêmico, médico, financeiro, jurídico e outros. Nesse contexto está compreendida a importância da classificação desses textos para a tomada de decisão.

A classificação de texto possibilita a organização de conjunto de documentos de texto de conteúdos variados por meio do processo de atribuir a cada documento de texto uma classe, categoria ou rótulo de acordo com o seu conteúdo. Esse processo é realizado de forma automática e a área de pesquisa que o investiga é a Mineração de Dados. Segundo Cichosz (2014), a Mineração de Dados é uma confluência entre as áreas de Aprendizado de Máquina e Estatística. O objetivo dessa área de pesquisa é transformar os dados em conhecimento e informação útil. Por meio dela, é possível a extração automática de padrões que representam o conhecimento implícito em grandes bases de dados (HAN et al., 2011; WITTEN et al., 2016).

A classificação é uma das principais tarefas da Mineração de Dados. Em um problema de classificação, utiliza-se um conjunto de dados (instâncias), já rotulados (com a informação de suas classes), para treinar um modelo. Pretende-se que esse modelo seja capaz de aprender com

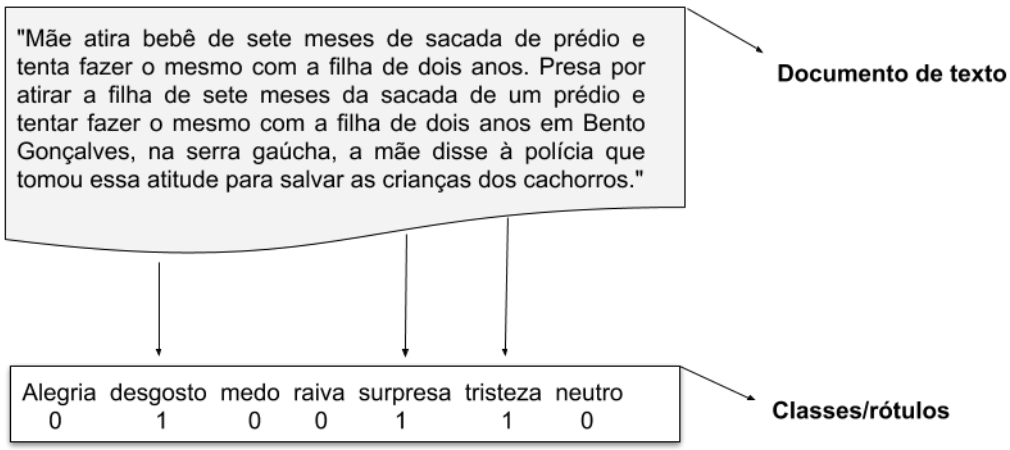
as características dos dados passados a ele e, a partir do aprendizado adquirido, seja capaz de classificar novos dados (WITTEN et al., 2016). Desta forma, na classificação de texto, modelos são treinados a partir de instâncias de textos com seus respectivos rótulos.

Na classificação monorrótulo, cada instância é associada a apenas uma classe. Ela é denominada classificação monorrótulo binária para um problema de classificação com apenas duas classes possíveis e classificação monorrótulo multiclasse, quando há mais de duas classes presentes no conjunto de dados (ASSIS, 2016).

No entanto, há problemas de classificação em que as instâncias podem estar associadas a mais de uma classe. Eles são denominados problemas de classificação multirrótulo. Como exemplo podem ser citadas a classificação das funções de proteínas, classificação de gêneros musicais e classificação de imagens. Nesse cenário, são aplicadas técnicas de classificação multirrótulo. A classificação de texto também pode ser um problema de classificação multirrótulo, em consequência de um texto poder estar associado a mais de uma classe (TSOUMAKAS; KATAKIS, 2007).

Um exemplo de classificação multirrótulo de um texto é apresentado por meio da Figura 1.1. Nesse exemplo, o documento de texto está associado às classes desgosto, surpresa e tristeza, concomitantemente. Tais classes referem-se à emoções de usuários ao lerem títulos de notícias. Nessa figura, 1 indica a associação do texto com a classe em questão e 0, o contrário.

Figura 1.1 – Exemplo de classificação multirrótulo de um texto.



Fonte: Elaborada pela autora (2020).

Vários métodos para a classificação multirrótulo têm sido propostos na literatura. No entanto, há pouco entendimento sobre qual método seria o mais apropriado para ser aplicado a um determinado problema (CHEKINA et al., 2011). Encontrar um algoritmo de classificação

que tenha um bom desempenho para um determinado conjunto de dados requer uma minuciosa investigação e muitos experimentos com classificadores (DO; NG, 2005), o que exige muito conhecimento das técnicas existentes e tempo, ou seja, escolher o classificador que melhor se adeque a um problema é um desafio.

Portanto, para atacar essa questão, novas abordagens de classificação têm incorporado meta-aprendizagem. Segundo Brazdil et al. (2009), a meta-aprendizagem é uma área de pesquisa em Mineração de Dados e Aprendizado de Máquina que possui objetivo duplo. Por um lado ela busca resolver, de forma automática, o problema de seleção das melhores técnicas disponíveis para o processo de aprendizagem, com objetivo de facilitar o trabalho do analista de dados. Por outro lado, ela preocupa-se em explorar o conhecimento adquirido por um modelo de aprendizagem em tarefas similares. Este trabalho possui como foco o primeiro objetivo supramencionado.

No cenário atual, com o aumento do volume e principalmente da variedade dos dados disponíveis, as técnicas de mineração de dados e aprendizado de máquina estão, cada vez mais, sendo empregadas nas mais diversas áreas. Com isso, a meta-aprendizagem desempenha um papel fundamental no emprego eficiente dessas técnicas. Brazdil e Giraud-Carrier (2018) apresentaram a atual conjuntura da área de meta-aprendizagem, em uma comparação entre a área de pesquisa no início dos anos 2000 até o ano de 2018. Nesse trabalho, os autores apontam os grandes avanços da área e enfatizam a importância da automatização dos processos de aprendizagem na atualidade.

1.1 Justificativa

Nos últimos anos, diversos estudos desenvolvidos na área de classificação têm focado em procedimentos que exigem esforços manuais para escolher o algoritmo mais apropriado para um determinado conjunto de dados (DO; NG, 2005). Contudo, com a variedade de algoritmos de classificação disponíveis na literatura e a diversidade dos dados para classificação, essa tarefa tornou-se muito árdua para os analistas humanos e, por isso, justifica-se a utilização de ferramentas automatizadas para a sua execução.

Um algoritmo de classificação pode ter um bom desempenho para alguns problemas de classificação, mas não em outros. Por esta razão, algumas abordagens de combinação de classificadores são geralmente utilizadas. Porém, essa combinação de classificadores não garante um

bom desempenho preditivo e, além disso, pode atrapalhar na interpretabilidade dos resultados (ZHANG et al., 2017).

Como alternativa para encontrar o classificador mais adequado para um conjunto de dados, pesquisas recentes têm aplicado a meta-aprendizagem com o objetivo de automatizar a seleção do(s) classificador(es) mais adequado(s) para um determinado problema. Alguns trabalhos da literatura já demonstraram a eficácia da utilização da meta-aprendizagem para a seleção de classificadores (CHEKINA et al., 2011; CRUZ et al., 2015; TRIPATHI et al., 2015; ZHANG et al., 2017). No entanto, não foram encontrados na literatura trabalhos que aplicaram a meta-aprendizagem especificamente para a classificação multirrótulo de textos com a seleção de classificadores sendo realizada para cada instância de texto, que é o foco deste trabalho.

Nesse contexto, a automatização do processo de seleção de algoritmos de classificação multirrótulo para a classificação de textos, não só contribui com a área de classificação de texto, mas também apresenta sua relevância para a indústria ao prover agilidade e eficiência no processo de seleção de classificadores.

1.2 Objetivos

Diante ao exposto na Subseção 1.1, a abordagem proposta neste trabalho tem o intuito de automatizar o processo de seleção do melhor modelo de classificação para uma determinada instância de texto. Dessa forma, o objetivo geral deste trabalho é aplicar a meta-aprendizagem na classificação de textos multirrótulo.

Como objetivos específicos tem-se:

- Desenvolver um meta-classificador para a classificação de texto multirrótulo que selecione o classificador mais apropriado baseado em cada instância de texto.
- Demonstrar que a abordagem proposta pode contribuir na seleção de classificadores no problema de domínio textual multirrótulo.

1.3 Abordagem proposta

Nesta dissertação propõe-se um método para a classificação de textos multirrótulo cujo funcionamento pode ser dividido em duas etapas. No método proposto, a etapa inicial é responsável pela realização do treinamento de vários modelos de classificação. Para isso, utiliza-se um conjunto de treinamento constituído por textos multirrótulo. Os modelos são avaliados por meio

de métricas para a avaliação multirrótulo e, para cada instância do conjunto de treinamento, eles são comparados para se encontrar o conjunto de modelos mais apropriado. Por conseguinte, os resultados obtidos nessa avaliação são armazenados.

Na segunda etapa, o método aplica o conhecimento adquirido na etapa inicial para classificar uma nova instância. Para isso, é realizado um cálculo de similaridade entre textos com o propósito de encontrar as instâncias do conjunto de treinamento mais similares à instância que se deseja classificar. Com as instâncias encontradas, o conjunto de classificadores mais apropriado para as mesmas (identificado na etapa anterior) é utilizado na classificação da nova instância.

O método proposto foi comparado com três métodos base usando duas bases de dados compostas por textos escritos no idioma português do Brasil e quatro bases de dados com textos escritos no idioma inglês. Os resultados mostram a eficiência do método proposto em diferentes domínios de texto e que, portanto, é uma boa opção para apoiar analistas em uma tarefa de classificação de texto multirrótulo.

1.4 Estrutura do texto

O restante do texto está organizado conforme descrito a seguir. O Capítulo 2 apresenta a fundamentação teórica com os principais conceitos relacionados a este trabalho. No Capítulo 3, alguns trabalhos relacionados ao tema desta dissertação são apresentados. Em seguida, no Capítulo 4, é apresentada a descrição do método proposto. Na sequência, os experimentos computacionais são reportados no Capítulo 5, com a avaliação do método proposto, resultados e discussão. Para finalizar, no Capítulo 6 são apresentadas as considerações finais englobando uma visão geral do método proposto, as contribuições da pesquisa e possíveis ideias para trabalhos futuros.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta os principais conceitos para a fundamentação teórica do trabalho. A Seção 2.1 introduz os tópicos Mineração de Dados e o Processo de Descoberta do Conhecimento em Bases de Dados com a descrição de suas etapas. Inclusa no processo de mineração, a Mineração de Texto (Seção 2.1.2) é apresentada englobando os conceitos de Processamento de Linguagem Natural (Seção 2.1.3) com algumas de suas técnicas (Seção 2.1.4), processo de classificação de texto (Seção 2.1.5), técnicas de classificação (Seção 2.1.8), avaliação de modelos de classificação (Seção 2.1.9) e classificação multirrótulo (Seção 2.1.10). Por fim, na Seção 2.2, é introduzido o conceito de Meta-aprendizagem com tópicos correlatos como aprendizagem-base e meta-conhecimento (Seção 2.2.1) e meta-atributos (Seção 2.2.2).

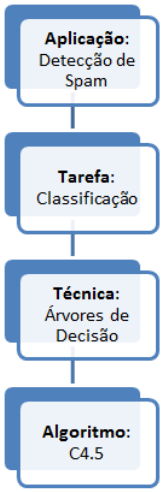
2.1 Mineração de Dados

A Mineração de Dados é uma confluência entre Aprendizado de Máquina e Estatística. Devido ao acelerado aumento dos dados disponíveis para análises e extração de conhecimento, a mineração de dados passou a ser destaque nas indústrias e no meio acadêmico (CICHOSZ, 2014). O objetivo da Mineração de Dados é transformar os dados em conhecimento e informação útil. Por meio dela, é possível a extração automática de padrões que representam o conhecimento implícito em grandes bases de dados (HAN et al., 2011; WITTEN et al., 2016).

Segundo Rokach e Maimon (2014), a Mineração de Dados vem sendo utilizada em diversas áreas e seus benefícios podem ser notados em importantes aplicações, tais como detecção de fraude, análise de crédito e planejamento de demanda. Para os autores, o processo de mineração pode ser organizado por meio de um modelo de quatro camadas (Figura 2.1), as quais são descritas a seguir:

- Aplicação: o problema a ser solucionado.
- Tarefa: as tarefas de Mineração de Dados, tais como classificação, regressão e agrupamento.
- Técnica: técnica a ser empregada, tais como árvores de decisão e redes neurais artificiais (RNAs).
- Algoritmo: algoritmo utilizado, tais como C4.5 para indução de árvores de decisão e *Multilayer Perceptron* para a construção de RNAs.

Figura 2.1 – Exemplo das quatro camadas do processo de mineração de dados.



Fonte: Elaborada pela autora (2020).

Essas camadas representam um conjunto de métodos aplicados em conjuntos de dados relacionados à camada de aplicação. O processo de mineração objetiva extrair regras e padrões desses dados (HAN et al., 2011). No entanto, para que os dados sejam minerados, eles podem ter que passar por uma série de tratamentos prévios. A literatura utiliza o termo processo de descoberta do conhecimento em bases de dados, do inglês *Knowledge Discovery from Data* (KDD), para descrever todas as etapas que reúnem desde a seleção dos dados para análises até a aplicação do conhecimento descoberto em todo o processo.

2.1.1 Etapas do processo de descoberta do conhecimento

A mineração de dados é a etapa principal do processo de KDD e, por isso, muitos pesquisadores acabam usando esses termos como sinônimos (HAN et al., 2011; ROKACH; MAIMON, 2014). Rokach e Maimon (2014) dividem o processo de descoberta do conhecimento em nove etapas:

1. **Desenvolvimento da compreensão do domínio da aplicação.** Nessa etapa inicial, são traçados os objetivos pretendidos no processo de descoberta do conhecimento. É na primeira etapa que busca-se compreender quais as técnicas devem ser utilizadas para o desenvolvimento do projeto. A compreensão do domínio da aplicação geralmente é adquirida por meio de uma revisão bibliográfica ou auxílio de um especialista.

2. **Obtenção de um conjunto de dados.** Com os objetivos definidos, o processo continua com a obtenção de um conjunto de dados do qual pretende-se extrair o conhecimento.

Para obter um conjunto de dados de qualidade, possíveis integrações de bases de dados são examinadas. Nessa etapa, também são selecionados os atributos do conjunto de dados que serão considerados durante o processo.

3. **Limpeza dos dados.** A limpeza dos dados faz parte do pré-processamento. Por meio dela, analisa-se se há a presença de dados incompletos, inconsistentes e ruídos. Segundo Bramer (2013), os ruídos são valores registrados de forma incorreta no conjunto de dados. Por exemplo, um valor numérico pode ser registrado com uma vírgula fora do lugar: 23,4 ao invés de 2,34. O pré-processamento geralmente é aplicado com o auxílio de ferramentas de mineração de dados, devido ao grande volume de dados em um conjunto.

4. **Transformação dos dados.** Nessa etapa, os dados são transformados para que possam ser minerados. De acordo com Han et al. (2011), algumas estratégias que podem ser utilizadas são:

   - Suavização: contribui para a remoção dos ruídos do conjunto de dados. Um exemplo de técnica de suavização envolve analisar os valores vizinhos do valor de um atributo e utilizar a média deles como valor padrão.
   - Agregação: processo de agregar os dados para reduzir o volume sem perder informação importante. Um exemplo dessa estratégia é transformar valores diários de vendas em valores mensais ou anuais.
   - Normalização: consiste em adotar uma escala para que os valores de todos os atributos fiquem dentro de uma mesma faixa.
   - Discretização: processo de transformar atributos contínuos (p. ex. idade) em intervalos discretos (p. ex. 0-16, 17-30, 31-50) ou em intervalos nominais (p. ex. criança, jovem, adulto).

   A transformação dos dados é crucial para se obter bons resultados na etapa de mineração.

5. **Escolha da tarefa de mineração de dados.** Com os dados pré-processados e transformados, o conjunto de dados está pronto para servir como entrada para os algoritmos de mineração. Nessa etapa, o analista deve escolher qual é a tarefa de mineração apropriada para o domínio da aplicação que vai responder aos objetivos traçados na etapa de desenvolvimento da compreensão do domínio da aplicação. De acordo com Rokach

e Maimon (2014), a Mineração de Dados possui duas categorias de tarefas principais: preditivas e descritivas. As tarefas preditivas aplicam-se às tarefas de Mineração de Dados supervisionadas, como a classificação e regressão, e as descritivas às tarefas de Mineração de Dados não supervisionadas, como o agrupamento e extração de regras de associação.

6. **Escolha do algoritmo de mineração.** Essa etapa consiste em encontrar o melhor algoritmo para realizar a mineração de dados. A literatura atual dispõe de uma variedade de técnicas de aprendizagem para a classificação, como as Máquinas de Vetores de Suporte, Redes Bayesianas, RNAs e outras.

7. **Aplicação do algoritmo escolhido.** O algoritmo escolhido é aplicado ao conjunto de dados. Nessa etapa, são realizadas várias execuções e ajustes nos parâmetros do algoritmo até encontrar resultados satisfatórios.

8. **Avaliação.** A penúltima etapa é para avaliar o desempenho do algoritmo e interpretar os padrões descobertos. Geralmente são empregadas várias métricas de avaliação para compreender a eficiência do modelo criado.

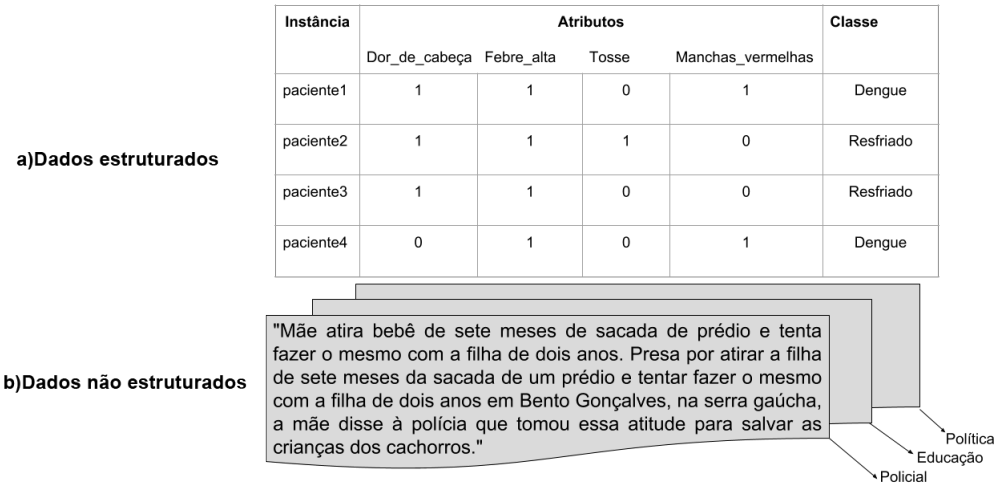
9. **Emprego do conhecimento descoberto.** A etapa final consiste em registrar e documentar os resultados. O modelo criado é utilizado para o desenvolvimento de algum sistema inteligente.

O processo de KDD pode ser aplicado em diversos tipos de dados. Trabalhos recentes têm aplicado o processo na mineração de dados textuais, o que gera a área denominada Mineração de Texto. A seção a seguir apresenta uma breve revisão sobre a Mineração de Texto.

2.1.2 Mineração de Texto

Para Weiss et al. (2015), a diferença entre Mineração de Dados e Mineração de Texto inicia-se na forma em que os dados são apresentados para a mineração (ver Figura 2.2). Na Mineração de Dados tradicional, os dados geralmente são estruturados em planilhas (ver Figura 2.2 a), enquanto na Mineração de Texto, os dados são apresentados em documentos de texto (ver Figura 2.2 b), que geralmente são descritos como informações não estruturadas. Dessa forma, os textos são preparados e estruturados por meio de técnicas da Mineração de Texto para tornarem-se adequados para as técnicas de Mineração de Dados tradicional.

Figura 2.2 – Dados estruturados vs. não estruturados.



Instância	Atributos				Classe
	Dor_de_cabeça	Febre_alta	Tosse	Manchas_vermelhas	
paciente1	1	1	0	1	Dengue
paciente2	1	1	1	0	Resfriado
paciente3	1	1	0	0	Resfriado
paciente4	0	1	0	1	Dengue

Fonte: Elaborada pela autora (2020).

A forma mais básica de representação de um texto é uma tabela com valores binários, onde a presença da palavra no texto é registrada com o valor 1, e sua ausência com o valor 0 (WEISS et al., 2015). A Tabela 2.1 apresenta um exemplo dessa transformação. Nessa tabela, as palavras representam os atributos do texto, que são as colunas da tabela, e cada linha da tabela é a representação de um documento de texto.

Tabela 2.1 – Representação do modelo básico de um texto para a mineração

Documentos de texto	Palavras												
	A	Mineração	de	Dados	é	O	objetivo	da	diferença	entre	Opinião	faz	parte
A Mineração de Dados é ...	1	1	1	1	1	0	0	0	0	0	0	0	0
O objetivo da Mineração de Dados é ...	0	1	1	1	1	1	1	1	0	0	0	0	0
A diferença entre Mineração de Dados ...	1	1	1	1	0	0	0	0	1	1	0	0	0
A Mineração de Opinião faz parte da ...	1	1	1	0	0	0	0	1	0	0	1	1	1

Fonte: Elaborada pela autora (2020).

Outras técnicas da Mineração de Texto atribuem pesos proporcionais à importância de cada palavra no texto. Dentre elas, a mais simples apenas contabiliza o número de vezes que a palavra aparece no texto e atribui o resultado ao valor do atributo. A Tabela 2.2 apresenta um exemplo dessa transformação.

Um conjunto de dados textuais para a mineração pode ser definido como uma coleção de documentos de texto. Portanto, para um domínio de classificação de interesses de usuários, os documentos da coleção devem denotar interesses, assim como para um domínio de análise de sentimentos os documentos da coleção devem denotar sentimentos.

Como observado nas Tabelas 2.1 e 2.2, o conjunto de palavras distintas de uma coleção de documentos de texto é utilizado para caracterizar os textos. Esse conjunto de palavras é

Tabela 2.2 – Representação de um texto com atribuição de pesos para as palavras

Documentos de texto	**Palavras**										
	A	Mineração	de	Dados	Os	dados	textuais	dois	requer	requerem	são
Os dados textuais são dados...	0	0	0	0	1	2	1	0	0	0	1
A Mineração de Dados requer dados...	1	1	1	1	0	1	0	0	1	0	0
Dados textuais requerem...	0	0	0	1	0	0	1	0	0	1	0
Dados dois dados...	0	0	0	1	0	1	0	1	0	0	0

Fonte: Elaborada pela autora (2020).

denominado dicionário. Uma série de tratamentos é aplicada nos textos para prepará-los para a mineração. As técnicas utilizadas no processo de preparação fazem parte do processamento de linguagem natural.

2.1.3 Processamento de Linguagem Natural

Linguagem é um conjunto de símbolos que expressam uma forma de comunicação, que pode ser oral, escrita ou outra. Embora alguns animais possuam vários sinais para se comunicarem, apenas os humanos conseguem, de forma natural, transmitir mensagens claras e concisas com o uso de sinais discretos. O Processamento de Linguagem Natural é um campo da Ciência da Computação e uma sub-área da Inteligência Artificial e da Linguística (CHOPRA et al., 2013), que foca na comunicação entre o ser humano e o computador e na aquisição de informação a partir da linguagem escrita (RUSSELL; NORVIG, 2010).

Diferentemente das linguagens formais (como as de programação), a linguagem natural não tem um modelo definido. Ela é ambígua e não apresenta um conjunto de sentenças único. Para conseguir extrair informações dessa linguagem, é necessária a utilização de técnicas que ajudem a melhor compreender as expressões de linguagem.

2.1.4 Técnicas de processamento de linguagem natural

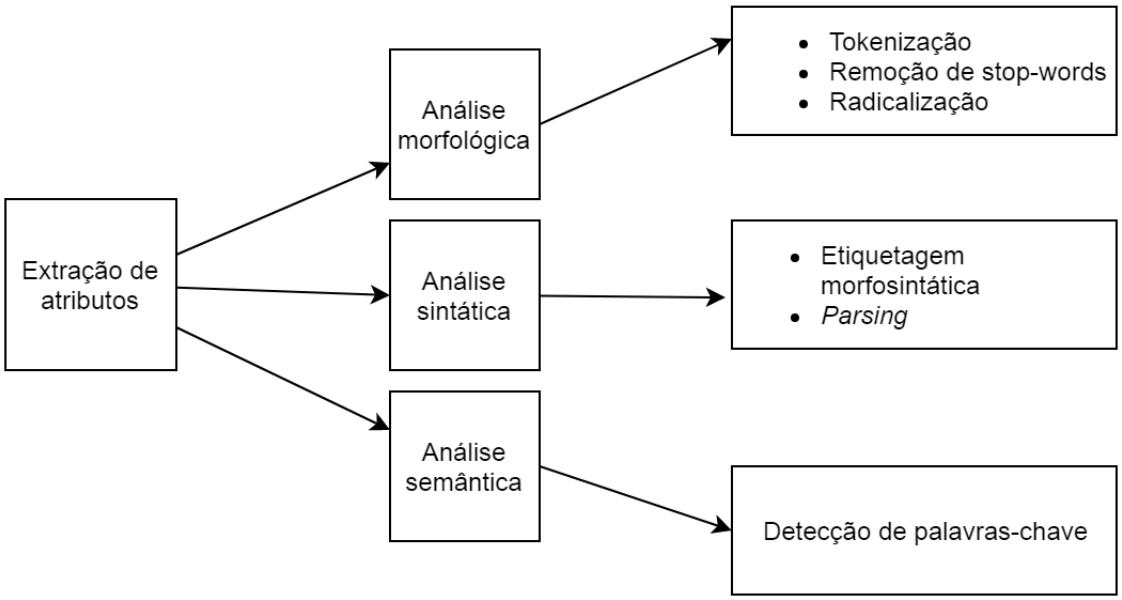
De acordo com Irfan et al. (2015), duas etapas básicas do Processamento de Linguagem Natural são aplicadas no pré-processamento de textos. São as seguintes:

- Extração de atributos.
- Seleção de atributos.

2.1.4.1 Extração de atributos

A extração de atributos é a etapa inicial do pré-processamento de textos. Ela compreende as análises morfológica, sintática e semântica (ver Figura 2.3).

Figura 2.3 – Categorias da extração de atributos.



Fonte: Adaptada de Irfan et al. (2015).

A morfologia, de acordo com a linguística, refere-se à formação de palavras. A análise morfológica é a identificação e descrição da estrutura das palavras (CHOPRA et al., 2013). Segundo Irfan et al. (2015), algumas das técnicas de extração de atributos associadas à análise morfológica são:

- Tokenização: separação das palavras de um texto seguindo uma delimitação pré-definida. O texto pode ser separado por palavras, frases ou outras partes importantes, definidas como *tokens*. As pontuações são, geralmente, removidas.

- Remoção de *stop-words*: removem-se todas as palavras consideradas irrelevantes para a classificação do texto. As classes de palavras geralmente removidas são os artigos, algumas conjunções e preposições.

- Radicalização: processo de reduzir a palavra para o seu radical. Por exemplo, modificar o verbo *sonhando* para seu radical *sonh*.

Além das etapas supramencionadas, outra etapa bastante utilizada no pré-processamento de texto, que envolve a estrutura da palavra, é a conversão do texto para letras minúsculas

(UYSAL; GUNAL, 2014). Também inclusa nessa categoria está a lematização, cujo objetivo é transformar uma palavra em sua forma base. Por exemplo, os verbos são transformados na forma infinitiva, o que remove a variedade de flexão resultante da conjugação nos tempos verbais. Para a lematizatição ser aplicada, é necessário realizar a etiquetagem morfosintática para identificar as classes gramaticais de cada termo da coleção (HOTHO et al., 2005).

A Tabela 2.3 mostra exemplos das técnicas de pré-processamento citadas anteriormente. O texto utilizado para exemplificar o uso das técnicas é: *"O objetivo da Mineração de Dados é transformar os dados em conhecimento"*.

Tabela 2.3 – Exemplos de aplicação das técnicas de pré-processamento da análise morfológica

Conversão para a letra minúscula	o objetivo da mineração de dados é transformar os dados em conhecimento
Tokenização	['O', 'objetivo', 'da', 'Mineração', 'de', 'Dados', 'é', 'transformar', 'os', 'dados', 'em', 'conhecimento']
Remoção de stop-words	objetivo Mineração Dados é transformar dados conhecimento
Radicalização	O objet da Miner de Dad é transform os dad em conhec
lematização	O objet da Miner de Dad ser transform o dad em conhec

Fonte: Elaborada pela autora (2020).

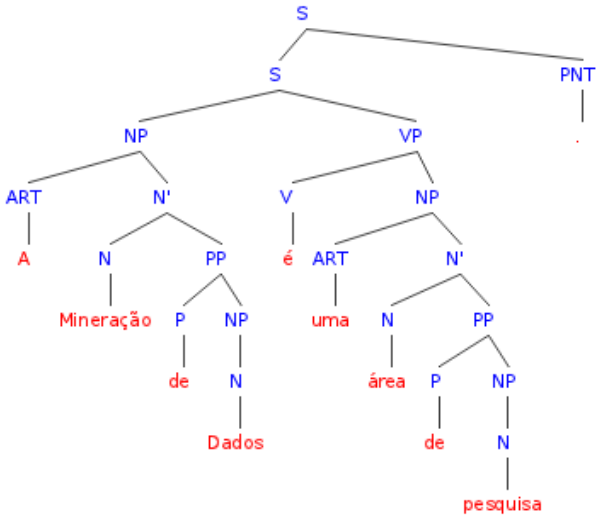
A análise sintática refere-se a estrutura das sentenças. Ela busca encontrar as relações entre as palavras de acordo com a estrutura gramatical de uma linguagem. Suas técnicas são:

- Etiquetagem morfosintática: utiliza informações gramaticais de cada palavra relacionadas ao contexto para adicionar conhecimento em uma sentença. Toda linguagem natural possui uma gramática com a classificação das palavras. A língua portuguesa utiliza 10 classes gramaticais: substantivo ou nome (N), artigo (ART), adjetivo (ADJ), pronome (PR), preposição (PREP), verbo (V), advérbio (ADV), numeral (NU), conjunção (C) e interjeição (I). Como exemplo, a frase *"A Mineração de Dados é uma área de pesquisa"* é etiquetada da seguinte forma: *"A_ART Mineração_N de_P Dados_N é_V uma_ART área_N de_C pesquisa_N"*.

- *Parsing*: examina a estrutura gramatical de uma sentença, ou seja, analisa a ordem gramatical em que as partes da sentença estão dispostas (IRFAN et al., 2015). Segundo Weiss et al. (2015), *parsing* é a forma mais sofisticada de processamento de texto, em que, por meio de uma única estrutura, geralmente uma árvore, as palavras de uma sentença são conectadas para se encontrar as relações entre elas. A Figura 2.4 representa a árvore criada para a frase *"A Mineração de Dados é uma área de pesquisa."*. Nessa figura, **S**(*sentence*) é a abreviação para sentença; **PNT**(*punctuation mark*) para sinal de pontuação; **NP**(*noun*

phrase) para frase nominal; **VP**(*verb phrase*) para frase verbal; **ART**(*article*) para artigo; **N**(*noun*) para substantivo; **PP**(*preposition phrase*) para frase prepositiva e **P**(*preposition*) para preposição.

Figura 2.4 – Exemplo de *parsing*.



Fonte: Silva et al. (2010).

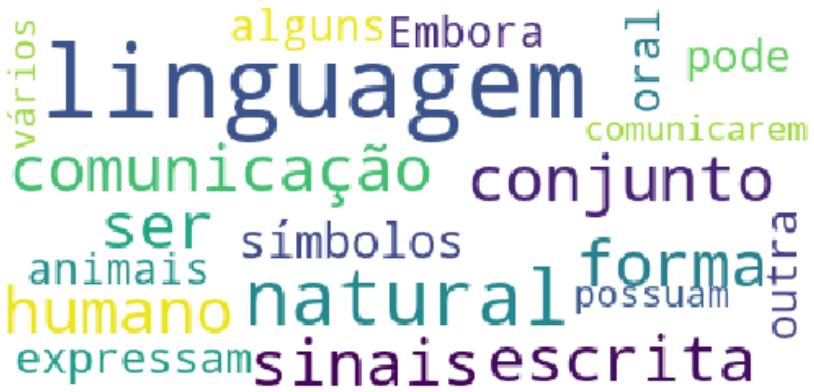
A semântica refere-se ao significado da sentença. A análise semântica permite a compreensão do significado da sentença por dicionário ou por extração do próprio contexto (CHOPRA et al., 2013). Um exemplo de técnica utilizada para extrair o significado de um texto é a detecção de palavras-chave (IRFAN et al., 2015).

A Figura 2.5 apresenta um exemplo de uma nuvem de palavras, do inglês *word cloud*, a qual é, geralmente, utilizada para visualizar as principais palavras de um texto. Nesse exemplo, foram selecionadas 20 palavras detectadas como palavras-chave do texto escrito na Subseção 2.1.3. O destaque no tamanho das palavras é dado pela relevância que elas possuem para representar todo o texto.

Outra técnica capaz de incluir o conteúdo semântico das palavras são os vetores de palavras. Por meio deles as palavras são dispostas em um espaço n-dimensional e as correlações entre elas são ponderadas. Esses vetores são treinados por meio de um substancial número de palavras e conseguem extrair conhecimento a partir delas (HARTMANN et al., 2017).

Um exemplo de vetores de palavras é o *Word2Vec*, o qual foi desenvolvido por Mikolov et al. (2013a). O *Word2Vec* utiliza uma rede neural para aprender as relaçoes entre as palavras. O método apresenta dois modelos denominados *continous bag of words* (CBOW) e *continuos skip-gram* (Skip-gram). No modelo CBOW, a rede neural recebe como entrada um conjunto

Figura 2.5 – Exemplo de detecção de palavras-chave.



Fonte: Elaborada pela autora (2020).

de palavras vizinhas de uma palavra específica e a rede prediz a palavra; já o modelo Skip-gram, a rede neural recebe como entrada uma palavra específica e a rede prediz sua vizinhança (MIKOLOV et al., 2013a).

2.1.4.2 Seleção de atributos

Outra etapa de pré-processamento de texto é a seleção de atributos (IRFAN et al., 2015). Por meio dela, selecionam-se os melhores atributos do conjunto de dados para eliminar informação irrelevante e reduzir a dimensão da representação dos dados. Uma abordagem conhecida de seleção de atributos é a seleção baseada na frequência das palavras. Segundo Baeza-Yates e Ribeiro-Neto (2011), essa etapa é importante, pois a dimensão do conjunto de atributos de um domínio pode tornar o processo de classificação impraticável para alguns classificadores ou aumentar demasiadamente o tempo de processamento. A seleção de atributo reduz a dimensão dos dados ao selecionar apenas um subconjunto representativo de atributos para caracterizar os documentos. Uma métrica comumente adotada nessa etapa é a *Term Frequency - Inverse Document Frequency* (TF-IDF).

A métrica TF-IDF permite realizar a seleção baseada na frequência dos termos nos textos. TF-IDF é um esquema de pesos que quantifica a importância dos termos para descrever um texto. De acordo com Baeza-Yates e Ribeiro-Neto (2011), é possível selecionar apenas os termos que apresentem os pesos TF-IDF mais altos para representar os textos. A Equação 2.1 apresenta uma das variantes do cálculo TF-IDF de um termo. Essa fórmula é utilizada se a frequência do termo no texto (TF) for maior que 0, caso contrário o peso atribuído é 0. Nessa equação, N representa o número total de documentos de texto na coleção e n a quantidade de

documentos de texto da coleção em que o termo aparece.

$$TF\text{-}IDF = (1 + logTF) \times log\frac{N}{n} \tag{2.1}$$

De acordo com Baeza-Yates e Ribeiro-Neto (2011), outras métricas muito aplicadas na seleção de atributos, são: *chi square*, informação mútua e ganho de informação.

- *Chi square* (X^2): por meio dessa métrica , seleciona-se um número pré-definido finito de palavras que melhor representam o texto. Nessa métrica, é atribuído um valor positivo ou negativo para cada palavra do texto representando a correlação com uma categoria particular C. Quanto mais alto o valor atribuído a palavra, maior a sua correlação com a categoria C.

- Informação mútua: é a medida de quanto a informação sobre um termo contribui para a correta classificação de uma classe. A informação mútua será zero se os termos forem totalmente independentes.

- Ganho de informação: é complementar a informaçao mútua. Ela não se importa apenas com a informação da presença do termo em um documento, mas também com sua ausência.

As técnicas de processamento de linguagem natural são aplicadas aos textos para prepará-los para as tarefas de mineração. A seção seguinte apresenta uma breve revisão sobre a classificação de texto.

2.1.5 Processo de classificação de texto

O processo de classificação é dividido em duas etapas: treinamento e teste (HAN et al., 2011). Na etapa de treinamento, utiliza-se uma amostra representativa de uma base de dados rotulada, conhecida como conjunto de treino, para treinar um modelo de classificação. O objetivo do treinamento é encontrar um classificador que identifique a correlação entre as características dos dados e suas classes (AGGARWAL; ZHAI, 2012; WITTEN et al., 2016). Na etapa de teste, o modelo treinado é avaliado utilizando-se uma porcentagem da base de dados rotulada, conhecida como conjunto de teste. Se a acurácia preditiva obtida durante a avaliação do modelo for satisfatória, então ele passa a ser utilizado para fazer a predição das classes de novas instâncias.

Os conjuntos de dados utilizados no processo de classificação são constituídos por um conjunto de atributos valorados. No caso da classificação de texto, como exemplos de atributos tem-se aqueles constituídos pelos próprios termos que ocorrem nos textos. Assim, a valoração é dada pela ocorrência dos termos nos textos. A Tabela 2.4 apresenta um conjunto de dados composto por cinco documentos, caracterizados por seus atributos, sendo o último o atributo classe. Os valores de cada atributo são apresentados na forma binária, em que 0 representa a ausência da palavra no texto e 1 a sua presença.

Tabela 2.4 – Exemplo de instâncias e atributos de um conjunto de dados.

Instâncias	**Atributos**														
	A	Mineração	de	Dados	é	O	objetivo	da	diferença	entre	dois	números	soma	...	**classe**
A diferença entre objetivo ...	1	0	0	0	0	0	1	0	1	1	0	0	0	...	psicologia
O objetivo da Mineração de Dados é ...	0	1	1	1	1	1	1	1	0	0	0	0	0	...	tecnologia
A diferença entre Mineração de Dados ...	1	1	1	1	0	0	0	0	1	1	0	1	0	...	tecnologia
Dados dois números ...	0	0	0	1	0	0	0	0	0	0	1	1	0	...	matemática
A soma entre números ...	1	0	0	0	0	0	0	0	0	1	0	1	1	...	matemática

Fonte: Elaborada pela autora (2020).

2.1.6 Similaridade entre textos

A similaridade do cosseno é uma das métricas mais utilizadas quando se quer encontrar a semelhança entre dois documentos. Ela é o resultado entre o produto interno de dois vetores pela normalização dos mesmos (ver Equação 2.2) (MANNING et al., 2008). Quanto maior for o valor retornado pela similaridade do cosseno, maior a similaridade entre os documentos.

Para exemplificar, se os vetores que representam dois documentos apontam para o mesmo ponto em um plano cartesiano bidimensional, significa que o ângulo formado por eles é de 0°. O cosseno de 0° é igual a 1, ou seja, a similaridade é total. Em contrapartida, se o ângulo formado por dois documentos nesse mesmo plano for 90°, a similaridade entre eles será nula, pois cosseno de 90° é igual a 0.

$$similaridade\text{-}cosseno(x,y) = \frac{xy^T}{||x||||y||} \tag{2.2}$$

em que:

x e y são vetores que representam documentos de textos.

2.1.7 Aplicações da classificação de texto

Enquanto as pesquisas em Recuperação de Informação focam em agilizar e falicitar o acesso à informação, a classificação de texto objetiva encontrar padrões nos textos que possam ser úteis para a tomada de decisão (AGGARWAL; ZHAI, 2012).

Em Aggarwal e Zhai (2012) são apresentadas quatro áreas em que a classificação de texto é geralmente aplicada, a saber:

- Organização e filtragem de artigos de notícias.
- Organização e recuperação de documentos.
- Mineração de opinião.
- Classificação de e-mail e filtragem de *spam*.

Algumas das técnicas de classificação que podem ser utilizadas para a classificação de texto e são descritas na seção a seguir.

2.1.8 Técnicas de classificação

Diferentes técnicas de classificação podem ser utilizadas na construção dos modelos de classificação. A seguir será apresentada uma breve revisão de algumas dessas técnicas, as quais foram utilizadas neste trabalho, a saber:

- Árvore de Decisão (Seção 2.1.6.1).
- Máquina de Vetor de Suporte (Seção 2.1.6.2).
- Técnicas Bayesianas (Seção 2.1.6.3).
- Redes Neurais Artificiais (Seção 2.1.6.4).

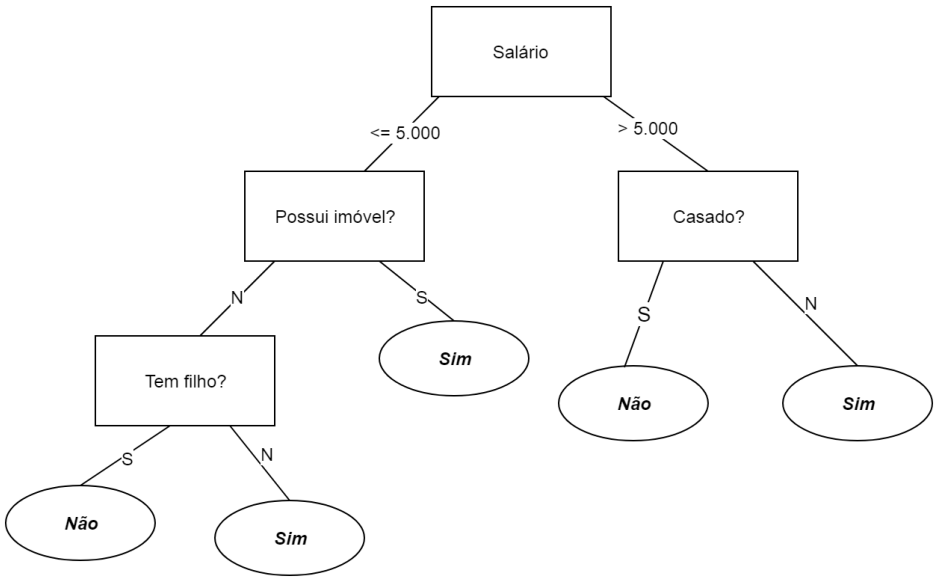
2.1.8.1 Árvore de decisão

As árvores de decisão são modelos computacionais conhecidos pela simplicidade na implementação e estão entre os mais utilizados em aprendizado de máquina (RUSSELL; NORVIG, 2010; DEAN, 2014). Segundo Rokach e Maimon (2014), um algoritmo de indução de árvore de decisão particiona o conjunto de dados de forma recursiva. A árvore é uma estrutura formada por um nó raiz, nós internos e nós folhas.

Em uma árvore de decisão, os nós raiz e internos representam os atributos, os ramos representam testes sobre os valores desses atributos e os nós folhas representam as classes. A Figura 2.6 mostra um exemplo de árvore de decisão para representar a decisão de concessão de crédito. Por meio das características de um cliente, a árvore de decisão prediz se ele vai conseguir o crédito ou não.

Na classificação, as árvores de decisão são utilizadas da seguinte forma: os valores dos atributos de uma instância de teste são testados na árvore de decisão por meio de um caminho em sua estrutura que inicia-se no nó raiz até um nó folha, onde contém a predição da classe da instância (HAN et al., 2011).

Figura 2.6 – Exemplo de uma árvore de decisão para representar a decisão de concessão de crédito.



Fonte: Elaborada pela autora (2020).

Alguns algoritmos populares para indução de árvores de decisão são: CART (BREIMAN et al., 1984), C4.5 (QUINLAN, 1993) e CHAID (KASS, 1980).

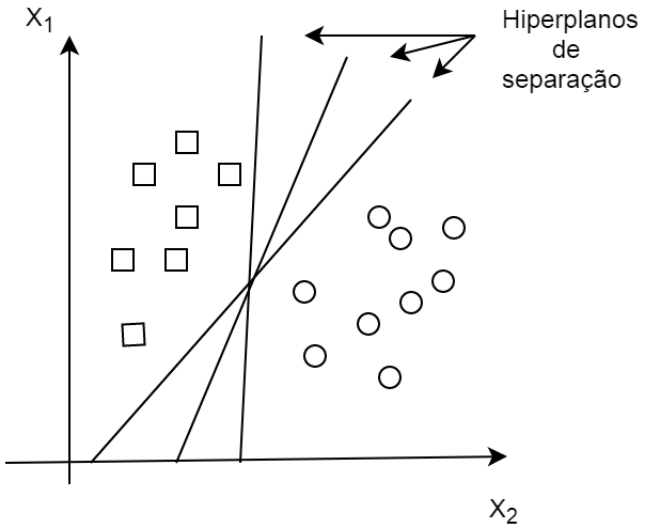
2.1.8.2 Máquina de vetor de suporte

Máquina de vetor de suporte, do inglês, *Support Vector Machine* (SVM), é uma técnica de classificação que foi proposta por volta da metade da década de 90, baseada na teoria do aprendizado estatístico, que busca encontrar funções preditivas a partir de dados utilizados como exemplos. Essa técnica caracteriza-se por apresentar algumas propriedades, tais como: agilidade no processo de aprendizado, obtenção de um conjunto de vetores de suporte ao cons-

truir a regra de decisão e uma solução única no problema de otimização para a sua construção (VAPNIK, 1999).

Uma SVM consegue classificar dados linearmente separáveis e não linearmente separáveis. O objetivo dessa técnica é criar um hiperplano que consiga com eficácia separar dados de entrada em duas classes, de tal maneira que quando um novo vetor de dados for inserido para teste, ele seja classificado corretamente (BELL, 2014). A Figura 2.7 representa um espaço bidimensional contendo instâncias associadas a duas classes distintas (quadrados e círculos). Nessa figura são apresentados três possíveis hiperplanos que separam as classes.

Figura 2.7 – Exemplo de uma máquina de vetor de suporte para um problema de classificação binário com três hiperplanos separando as duas classes.

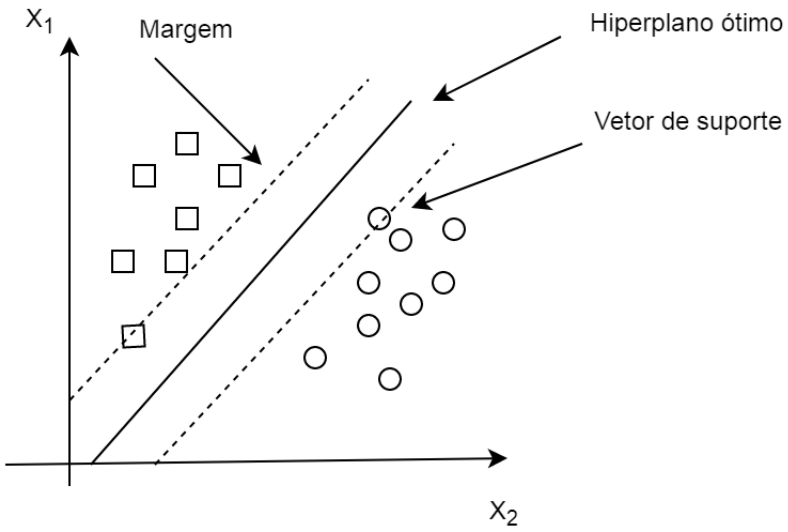


Fonte: Adaptada de Bell (2014).

Para que seja encontrado o hiperplano ótimo, aquele que representa a maior separação entre as classes, é necessária a criação de uma margem máxima entre os exemplos de treinamento, vetores de entrada, e o hiperplano. Os vetores que determinam a distância da margem e o hiperplano ótimo são considerados os vetores de suporte (BELL, 2014). A Figura 2.8 representa um hiperplano ótimo para as instâncias plotadas em um espaço bidimensional.

Os exemplos de treinamento que são os vetores de suporte são de suma importância para o classificador SVM. No entanto, diferente de alguns classificadores em que a complexidade é determinada pela dimensionalidade dos dados, em uma SVM a complexidade é caracterizada pelo número de vetores de suporte (VAPNIK, 1999; HAN et al., 2011).

Figura 2.8 – Exemplo de uma máquina de vetor de suporte para um problema binário com um hiperplano ótimo separando as duas classes.



Fonte: Adaptada de Bell (2014).

2.1.8.3 Técnicas Bayesianas

De acordo com Han et al. (2011), as técnicas Bayesianas são muito utilizadas para a classificação de texto. Elas são métodos estatísticos que predizem a probabilidade de uma instância pertencer a uma classe. Essas técnicas são baseadas no Teorema de Bayes, que consiste em fornecer a probabilidade da ocorrência de um evento dado algum conhecimento prévio. Assim, na classificação, o Teorema de Bayes pode ser representado por meio da Equação 2.3, em que $P(c|x)$ representa a probabilidade de uma instância (x) pertencer a uma classe (c).

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)} \tag{2.3}$$

Os classificadores *Multinomial Naive Bayes* (MNB) e *Bernoulli Naive Bayes* (BNB) são exemplos de classificadores Bayesianos. O classificador *Multinomial Naive Bayes* (MNB) foi proposto para a classificação de texto. Ele corresponde a uma evolução do classificador *Bernoulli Naive Bayes* (BNB), que é um classificador para texto que considera cada palavra como uma variável booleana. O MNB leva em consideração a informação do número de vezes que uma palavra ocorre em um documento, conforme apresentado por Zhang et al. (2017).

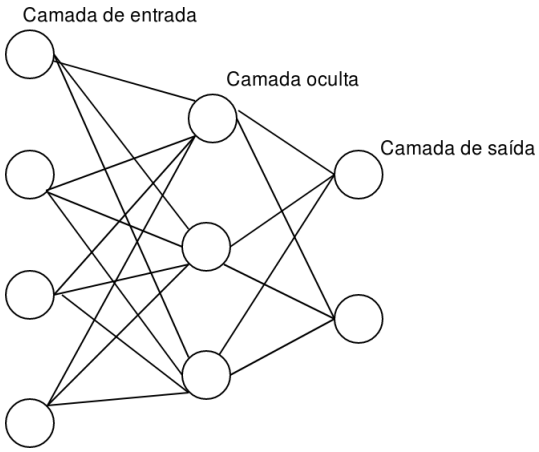
Alguns classificadores Bayesianos assumem independência entre os atributos do conjunto de dados. Desse modo, em problemas de classificação em que os atributos são correlacionados, eles podem não apresentar bons desempenhos na classificação.

2.1.8.4 Redes neurais artificiais

As Redes Neurais Artificiais(RNAs) são modelos computacionais que tem como fundamentação o processo de aquisição de conhecimento que ocorre no cérebro (VAPNIK, 1999). A estrutura de uma RNA pode ser definida como um conjunto de nós conectados entre si, em que cada conexão contém um peso numérico associado a ela. Esse peso indica a força e o sinal da conexão (RUSSELL; NORVIG, 2010).

De acordo com Bell (2014), uma rede neural é baseada em entradas e saídas e o componente base das RNAs é o *perceptron*, o qual recebe um sinal de entrada e, por meio de alguma função, retorna uma saída. Os *perceptrons* podem ser de uma ou múltiplas camadas. Os de uma camada são aplicados em problemas linearmente separáveis, enquanto os de múltiplas camadas envolvem camadas ocultas e são utilizados para problemas mais complexos (não linearmente separáveis). A Figura 2.9 representa um *perceptron* de múltiplas camadas com uma camada oculta.

Figura 2.9 – Exemplo de um *perceptron* de múltiplas camadas com uma camada oculta.



Fonte: Adaptada de Bell (2014).

Em uma RNA, a camada de entrada recebe os valores dos atributos provenientes dos dados de treinamento. Em seguida, conectados à ela está a camada oculta ou escondida, a qual é responsável por realizar o processamento das informações de acordo com seus pesos. Para finalizar o processo, a camada de saída retorna o resultado do algoritmo conforme suas entradas (FERREIRA, 2014).

2.1.9 Avaliação de modelos de classificação

A avaliação de modelos de classificação objetiva analisar o desempenho do modelo criado para a predição. As diferentes métricas de avaliação são utilizadas de acordo com o problema a ser analisado (CICHOSZ, 2014). Segundo Baeza-Yates e Ribeiro-Neto (2011), a avaliação é um ponto determinante para a validação do modelo proposto. Métricas tradicionalmente utilizadas na avaliação são: Precisão, Revocação, F1, Macro F1 e Micro F1 (ou Acurácia).

As métricas Precisão (P) e Revocação (R) de uma classe c são representadas por meio das Equações 2.4 e 2.5, respectivamente. Nessas equações, VP_c (verdadeiros positivos) representa o número de instâncias da classe c corretamente classificadas pelo modelo, FP_c (falsos positivos) representa o número de instâncias que não pertencem a classe c, mas que foram incorretamente atribuídas à classe c pelo modelo e FN_c (falsos negativos) representa o número de instâncias que pertencem a classe c mas que foram incorretamente classificadas pelo modelo. Para Han et al. (2011), a Precisão pode ser pensada como uma medida de exatidão e a Revocação uma medida de completude.

$$P_c = \frac{VP_c}{VP_c + FP_c} \tag{2.4}$$

$$R_c = \frac{VP_c}{VP_c + FN_c} \tag{2.5}$$

A F1 é uma métrica que combina a Precisão (P) e a Revocação (R) e funciona como um balanceamento da importância das duas. A Equação 2.6 mostra como a F1 é computada.

$$F1_c = \frac{2 \times P_c \times R_c}{P_c + R_c} \tag{2.6}$$

A Macro F1, de acordo com Baeza-Yates e Ribeiro-Neto (2011), considera a importância de cada classe do conjunto de dados. Por isso, ela é a razão entre o somatório das F1 de cada classe e o número total de classes ($|C|$) do conjunto de dados. A métrica Macro F1 é representada por meio da Equação 2.7.

$$Macro\ F1 = \frac{\sum_{i=1}^{|C|} F1c_i}{|C|} \tag{2.7}$$

A métrica Micro F1 atribui a cada exemplo do conjunto de dados a mesma importância. A Equação 2.8 mostra como a Micro F1 é calculada. É importante observar que ela utiliza a

Precisão e Revocação sobre todas as classes. A Equação 2.9 apresenta o cálculo da Precisão sobre todas as classes e a Equação 2.10 representa o cálculo da Revocação sobre todas as classes.

$$Micro\ F1 = \frac{2 \times P \times R}{P + R}, \tag{2.8}$$

em que:

$$P = \frac{\sum_{c_i \in C} VP_c}{\sum_{c_i \in C} (VP_c + FP_c)} \tag{2.9}$$

e

$$R = \frac{\sum_{c_i \in C} VP_c}{\sum_{c_i \in C} (VP_c + FN_c)} \tag{2.10}$$

2.1.10 Classificação multirrótulo

Diferentemente da tradicional classificação monorrótulo, onde cada instância está associada a apenas uma classe, em problemas de classificação multirrótulo, uma instância pode estar associada a mais de uma classe (ZHANG; ZHOU, 2014; READ et al., 2016). Por exemplo, considerando uma coleção de textos sobre filmes, na classificação multirrótulo, um texto pode estar associado a mais de uma classe (p. ex., ação, comédia, romance). A Figura 2.10 apresenta um exemplo de uma base de dados utilizada em classificação multirrótulo.

Figura 2.10 – Exemplo de classificação multirrótulo.

Instância	Classe
doc_0	drama, suspense
doc_1	ação, comédia, romance
doc_2	suspense, drama, romance
doc_3	romance, drama
doc_4	comédia

Fonte: Elaborada pela autora (2020).

Como demonstrado na Figura 2.10, um problema de classificação multirrótulo apresenta características peculiares. Como exemplos têm-se a cardinalidade e a densidade, ambas referentes à quantidade de rótulos presentes no conjunto de dados. Tais características fornecem informações relevantes sobre o conjunto de dados e estão intrinsicamente ligadas ao desempenho dos classificadores.

A cardinalidade dos rótulos fornece o número médio de rótulos por instância de um conjunto de dados. Seu cálculo é demonstrado por meio da Equação 2.11. Já a densidade dos rótulos, apresentada na Equação 2.12, é o resultado da cardinalidade pelo número total de

rótulos presentes no conjunto de dados (TSOUMAKAS; KATAKIS, 2007).

$$C = \frac{1}{N}\sum_{i=1}^{N}|Y_i| \tag{2.11}$$

em que:

N representa o número de instâncias presentes no conjunto de dados e $|Y_i|$ representa o número de rótulos para uma dada instância i.

$$D = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i|}{|L|} \tag{2.12}$$

em que:

N representa o número de instâncias presentes no conjunto de dados, $|Y_i|$ representa o número de rótulos para uma dada instância i e $|L|$ representa o número total de rótulos presente no conjunto de dados.

Os métodos utilizados na classificação multirrótulo são divididos em duas categorias principais: métodos de transformação do problema e métodos de adaptação de algoritmos (ZHANG; ZHOU, 2014). Nas subseções seguintes será apresentada uma breve descrição dessas duas categorias com exemplos de alguns dos seus métodos.

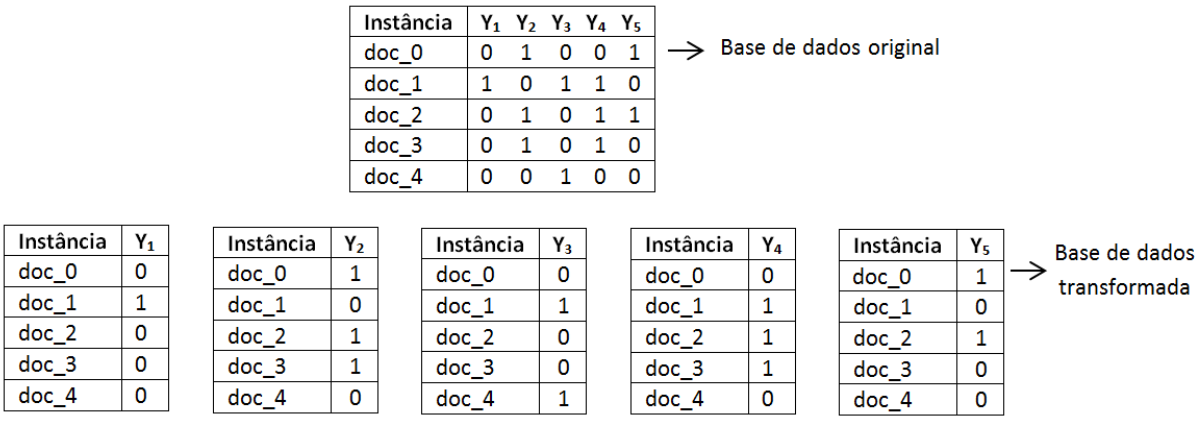
2.1.10.1 Métodos de transformação do problema

Nessa categoria, o problema de classificação multirrótulo é transformado em vários problemas monorrótulos que permitem a utilização de algoritmos de classificação tradicionais (para o contexto monorrótulo). Os principais métodos dessa categoria são: Relevância Binária, *Label Powerset* (LP) e Cadeia de Classificadores.

Na relevância binária, o problema multirrótulo é transformado em vários problemas de classificação binária e, a partir disso, são utilizados classificadores monorrótulo para cada um desses problemas. Apesar de ser uma técnica muito utilizada na classificação multirrótulo, uma de suas principais desvantagens é não levar em consideração a correlação entre rótulos existente no problema original. Na fase de aprendizagem, cada classificador binário, responsável por um único rótulo, trabalha de forma independente ignorando os demais rótulos (ZHANG et al., 2018). A classificação de uma nova instância é obtida a partir da união dos resultados dos classificadores individuais (TSOUMAKAS; KATAKIS, 2007).

A Figura 2.11 ilustra a transformação do problema multirrótulo a partir da base de dados apresentada na Figura 2.10 utilizando-se a relevância binária. A parte superior dessa figura apresenta a base de dados original e a parte inferior as bases obtidas após a transformação. Cada uma dessas bases transformadas será utilizada no treinamento de um classificador monorrótulo. A primeira coluna da figura representa as instâncias e a variável **Y** os rótulos aos quais elas estão associadas. Assim, 1 indica a associação de uma instância com um rótulo e 0, o contrário. Os rótulos Y_1, Y_2, Y_3, Y_4 e Y_5 correspondem, respectivamente, às seguintes classes: ação, drama, comédia, romance e suspense.

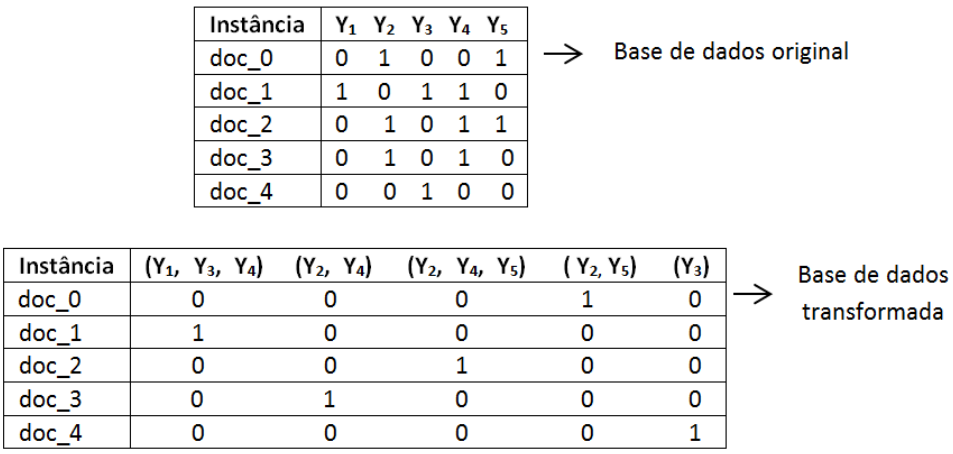
Figura 2.11 – Exemplo do método de transformação de um problema multirrótulo - Relevância Binária.



Fonte: Elaborada pela autora (2020).

Por meio do método *Label Powerset* (LP), o problema multirrótulo é transformado em um problema monorrótulo multiclasse a partir do mapeamento do conjunto de rótulos de cada instância em um único rótulo.

Figura 2.12 – Exemplo do método de transformação de um problema multirrótulo - *Label Powerset*.

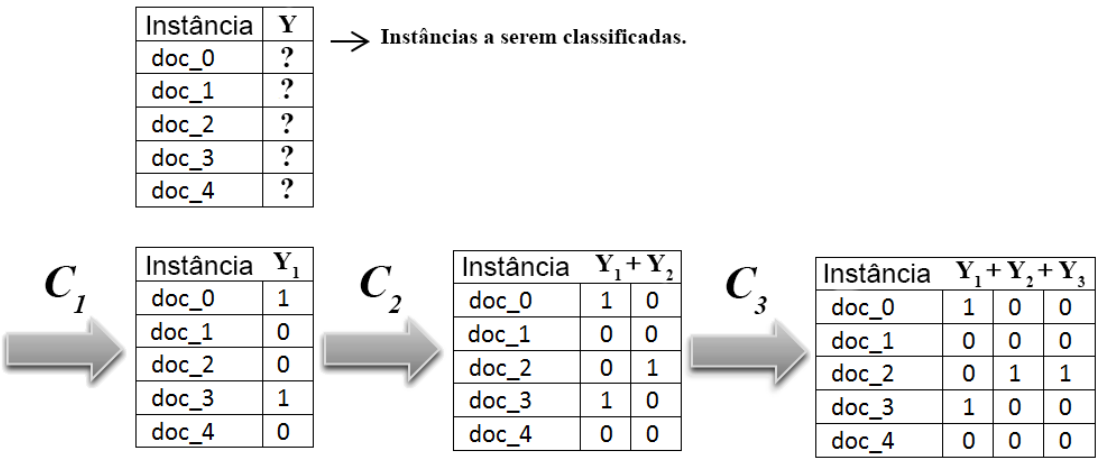


Fonte: Elaborada pela autora (2020).

Dessa forma, os algoritmos de aprendizagem monorrótulo passam a ser utilizados. Essa abordagem de transformação aumenta a complexidade do problema em razão do número de diferentes conjuntos de rótulos possíveis em uma base de dados, que pode chegar a 2^Y conjuntos, sendo Y o número de rótulos contidos no conjunto de dados original. Outro problema desse método é que ele reduz a quantidade de exemplos por classe (TSOUMAKAS; KATAKIS, 2007). A Figura 2.12 é uma representação da transformação da base de dados apresentada na Figura 2.10 utilizando-se o método *Label Powerset*.

Dada a limitação da relevância binária por não considerar a correlação entre os rótulos, algumas outras técnicas foram propostas na literatura para minimizar esse problema. Uma dessas técnicas é a cadeia de classificadores, que também cria um classificador binário para cada rótulo do conjunto de dados. No entanto, cada um desses classificadores, que são dispostos em uma cadeia, realiza a predição para um rótulo e o resultado obtido é utilizado como um novo atributo para o classificador seguinte da cadeia. A Figura 2.13 apresenta um problema de classificação multirrótulo e a representação de uma cadeia de classificadores composta pelo conjunto de classificadores $C = \{C_1, C_2, C_3\}$. Nessa figura, as instâncias a serem classificadas são passadas para o primeiro classificador da cadeia, o C_1, o qual realiza a predição para o primeiro rótulo, que será utilizada pelo segundo classificador da cadeia, o C_2. Em seguida, o classificador C_2 realiza a predição para o segundo rótulo utilizando os mesmos atributos utilizados por C_1 somado ao atributo criado a partir do resultado da predição de C_1. Da mesma forma, os resultados das predições realizadas por C_1 e C_2 são utilizados como atributos adicionais para a predição a ser realizada pelo próximo classificador da cadeia, o C_3. O classificador C_3, por sua vez, realiza a predição do terceiro rótulo.

Figura 2.13 – Exemplo do método Cadeia de Classificadores.



Fonte: Elaborada pela autora (2020).

Essa abordagem minimiza a desvantagem do método relevância binária e apresenta uma complexidade computacional aceitável (READ et al., 2011). No entanto, nessa abordagem, o desempenho do modelo está fortemente ligado à ordem com que os classificadores são dispostos na cadeia (SILVA, 2014).

2.1.10.2 Métodos de adaptação de algoritmos

Nessa categoria, os algoritmos monorrótulo são adaptados para, diretamente, serem aplicados ao problema multirrótulo. Como exemplos de adaptações têm-se: MLC4.5 (árvore de decisão multirrótulo) e ML-kNN (k-NN adaptado para o contexto multirrótulo).

O algoritmo MLC4.5, inicialmente proposto em Clare e King (2001) para lidar com os desafios de um conjunto de dados multirrótulo proveniente da área de Ciências Biológicas, é uma modificação do algoritmo de árvore de decisão C4.5. Uma árvore de decisão é construída seguindo a abordagem *top-down*, ou seja, o processo de criação do modelo inicia-se pelo topo da árvore, nó raiz, a partir da escolha de um atributo para representar aquele nó. A partir da escolha do atributo que representará o nó raiz, a base de dados é subdividida de acordo com os valores do mesmo e, para cada ramo criado a partir dessa subdivisão, o processo de escolha dos atributos que irão compor os novos nós da árvore se repete.

O algoritmo C4.5 utiliza a medida denominada entropia no processo de escolha do melhor atributo para representar cada nó da árvore. A Equação 2.13 ilustra o cálculo da entropia utilizado pelo C4.5. Nessa equação, a variável S representa o conjunto de instâncias de treinamento; $p(c_i)$ a probabilidade de uma dada classe i ocorrer em S e N é o número de classes presentes em S.

$$entropia(S) = -\sum_{i=1}^{N} p(c_i) log(p(c_i)) \tag{2.13}$$

Para que o algoritmo pudesse lidar com exemplos associados a mais de uma classe, o cálculo de entropia no MLC4.5 foi modificado, conforme a Equação 2.14. Nesse caso, cada vez que é calculada a probabilidade de uma classe ($p(c_i)$) em S, também é considerado no cálculo o valor do seu complemento, representado por $q(c_i)$.

$$entropia(S) = -\sum_{i=1}^{N} ((p(c_i) log(p(c_i)) + (q(c_i) log(q(c_i))) \tag{2.14}$$

Assim como no cálculo da entropia, o MLC4.5 realiza alterações para que as folhas da árvore possam retornar um conjunto de rótulos, ao invés de apenas um, como ocorre no contexto monorrótulo.

O algoritmo ML-kNN, apresentado em Zhang e Zhou (2005), é uma adaptação do *k-Nearest Neighbor*(k-NN). O k-NN realiza a classificação a partir de uma análise de similaridade entre a instância a ser classificada e aquelas existentes na base de dados de treinamento. Dessa forma, ele realiza a classificação sem a criação prévia de um modelo. Mais especificamente, ao receber uma nova instância para ser classificada, o k-NN busca os seus K vizinhos mais próximos na base de dados de treinamento e atribui à essa nova instância a classe mais frequente entre esses vizinhos (BAEZA-YATES; RIBEIRO-NETO, 2011).

Na sua versão adaptada para o contexto multirrótulo, o ML-kNN trabalha de forma independente com cada um dos rótulos. Dada uma instância para ser classificada, primeiro são identificados os seus k vizinhos mais próximos na base de dados de treinamento e, de acordo com o conjunto de rótulos dos mesmos, é predito o conjunto de rótulos da nova instância aplicando-se inferência bayesiana. Uma outra característica importante do ML-kNN é a capacidade de retornar um ranqueamento dos rótulos (ZHANG; ZHOU, 2005; TSOUMAKAS; KATAKIS, 2007).

Há também algoritmos para a classificação monorrótulo que podem ser diretamente aplicados a classificação multirrótulo. Um exemplo desses é o classificador *Multilayer Perceptron* (MLP), das redes neurais artificiais (ver Seção 2.1.8.4).

A literatura dispõe de um grande número de técnicas para classificação multirrótulo, desde aquelas que utilizam abordagens como as apresentadas neste trabalho até técnicas de combinação de classificadores. Exemplos dessas técnicas são encontrados em Tsoumakas e Vlahavas (2007), Read et al. (2008), Tsoumakas et al. (2008), Tenenboim-Chekina et al. (2010), Read et al. (2011).

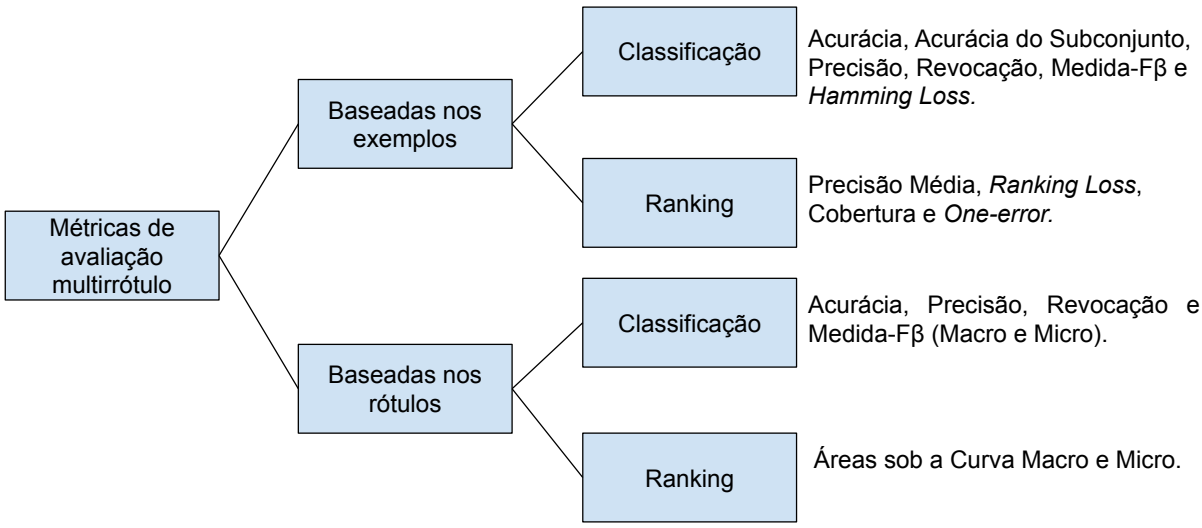
2.1.11 Avaliação dos modelos de classificação multirrótulo

Como na classificação multirrótulo cada exemplo pode estar associado a mais de uma classe, a avaliação desses modelos é mais complexa do que na classificação monorrótulo. Na classificação multirrótulo, as métricas de avaliação são divididas em duas categorias: baseadas em exemplos e baseadas em rótulos. A principal diferença entre elas é como o desempenho do classificador é calculado. Métricas baseadas em exemplos retornam uma média das diferenças

entre o conjunto verdadeiro e previsto de rótulos em todos os exemplos de teste, enquanto as métricas baseadas em rótulos são calculadas para cada rótulo e o desempenho do classificador é fornecido pelo valor médio considerando-se todos os rótulos (ZHANG; ZHOU, 2014). A Figura 2.14 apresenta essas categorias e seus exemplos.

As baseadas em exemplos são categorizadas em métricas de classificação e métricas de ranking. As métricas de classificação incluem: Acurácia, Acurácia do Subconjunto, Precisão, Revocação, Medida-F$^{\beta}$ e *Hamming Loss*. Já as métricas de ranking são: Precisão Média, *Ranking Loss*, Cobertura e *One-error*. De forma similar, as métricas baseadas em rótulos também são categorizadas em classificação e ranking. As métricas de classificação são: Acurácia, Precisão, Revocação, Medida-F$^{\beta}$ Macro e Medida-F$^{\beta}$ Micro. Já as métricas de ranking compreendem as Áreas sob a Curva Macro e Micro (ZHANG; ZHOU, 2014), como é apresentado na Figura 2.14.

Figura 2.14 – Métricas para avaliação dos modelos de classificação multirrótulo.



Fonte: Adaptada de Zhang e Zhou (2014).

Conforme um estudo realizado por Pereira et al. (2018), em trabalhos envolvendo classificação multirrótulo, as métricas de avaliação geralmente são escolhidas de forma arbitrária. Essa estratégia de escolha pode não ser a ideal, uma vez que muitas dessas métricas estão correlacionadas. De acordo com a análise de correlação entre as métricas realizada nesse trabalho, os autores sugerem a utilização das seguintes métricas para avaliação da classificação multirrótulo: *Hamming Loss*, *Ranking Loss* e/ou Cobertura e alguma das demais métricas (as quais são fortemente correlacionadas). Portanto, serão brevemente descritas aqui as métricas *Hamming Loss* e a Medida-F$^{\beta}$, dado que elas estão entre aquelas utilizadas neste trabalho.

A métrica *Hamming Loss*, a mais utilizada para a avaliação de classificadores multirrótulo, considera o erro na predição. Portanto, quanto mais seu valor aproximar-se de 0, melhor o desempenho do classificador. A Equação 2.15 apresenta o cálculo dessa métrica. Nessa equação, N representa o número de instâncias no conjunto de dados; Y_i, o conjunto de rótulos reais para uma dada instância i; P_i, o conjunto de rótulos preditos para a instância i e L, o conjunto total de rótulos presentes no conjunto de dados.

$$Hamming\text{-}loss = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i \neq P_i|}{|L|} \tag{2.15}$$

A Medida-F$^{\beta}$ é a média harmônica entre a Precisão (Equação 2.16) e a Revocação (Equação 2.17) do classificador, as quais também foram adaptadas para o problema multirrótulo para levar em conta acertos parciais na predição dos rótulos. A Equação 2.18 demonstra como é realizado o cálculo da Medida-F$^{\beta}$ com β = 1. Nessas equações, N representa o número de instâncias no conjunto de dados; Y_i, o conjunto de rótulos reais para uma dada instância i e P_i, o conjunto de rótulos preditos para a instância i.

$$\text{Precisão} = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i \cap P_i|}{|P_i|} \tag{2.16}$$

$$\text{Revocação} = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i \cap P_i|}{|Y_i|} \tag{2.17}$$

$$Medida\text{-}F = \frac{1}{N}\sum_{i=1}^{N}\frac{2|Y_i \cap P_i|}{|P_i| + |Y_i|} \tag{2.18}$$

Já a Medida-F$^{\beta}$ Micro, muito utilizada na avaliação de classificadores com β = 1, é baseada nos rótulos. Essa métrica aplica o somatório dos valores, falsos positivos (FP), falsos negativos (FN), verdadeiros positivos (VP) e verdadeiros negativos (VN), apresentados por cada rótulo. A Micro F1 é indicada principalmente para conjuntos de dados desbalanceados por realizar o cálculo de forma global. A Equação 2.19 e Equação 2.20 apresentam como essa métrica é calculada.

$$Micro\ F1 = \frac{2VP}{2VP + FP + FN} \tag{2.19}$$

Em que:

$$VP = \left(\sum_{i=1}^{|L|} VP_i\right), FP = \left(\sum_{i=1}^{|L|} FP_i\right), VN = \left(\sum_{i=1}^{|L|} VN_i\right), FN = \left(\sum_{i=1}^{|L|} FN_i\right) \tag{2.20}$$

Na seção a seguir é apresentado o conceito de meta-aprendizagem bem como suas ramificações.

2.2 Meta-aprendizagem

O termo meta-aprendizagem passou a ser utilizado na área de Aprendizado de Máquina a partir do ano 1990. Advinda da Psicologia, a meta-aprendizagem é a capacidade de um indivíduo descobrir as melhores formas de aquisição do seu aprendizado e aplicá-las para seu próprio aprendizado sem interferências de um instrutor. Na Mineração de Dados e no Aprendizado de Máquina, devido ao grande número de algoritmos de aprendizagem com características diferentes e várias formas de pré-processamento dos dados para análises, a meta-aprendizagem contribui para que, de forma automática, sejam selecionados os melhores conjuntos de técnicas de aprendizagem para determinados problemas de classificação (LEMKE et al., 2015).

Segundo Rokach e Maimon (2014), a meta-aprendizagem procura explicar os diferentes desempenhos de um algoritmo de aprendizagem quando aplicado a um determinado problema. Por isso, ela busca caminhos para compreender e direcionar o algoritmo mais apropriado para solucionar o problema. Como um exemplo, em um trabalho realizado por Chekina et al. (2011), o processo de meta-aprendizagem pôde ser definido nas seguintes etapas:

- Avaliação do desempenho dos métodos investigados em diversos conjuntos de dados;
- Aquisição das características descritivas de cada conjunto de dados;
- Criação de um meta-conjunto de dados com as características adquiridas;
- Construção de um meta-classificador a partir do meta-conjunto de dados para direcionar um conjunto de dados ao classificador mais apropriado.

Uma outra forma de meta-aprendizagem é denominada transferência indutiva (do inglês *inductive transfer*). Muitas pesquisas nessa área focam em desenvolver um método individual de aprendizagem e não um sistema de aprendizagem. Nessa abordagem, são investigadas tarefas de aprendizagem relacionadas e o conhecimento adquirido em alguns domínios é transferido

para outros (LEMKE et al., 2015). Um exemplo dessa abordagem é descrito no trabalho desenvolvido por Do e Ng (2005).

A abordagem investigada neste trabalho, assim como apresentado em Chekina et al. (2011), possui o objetivo de selecionar o classificador mais apropriado para um conjunto de dados. Contudo, nesta abordagem, o foco é selecionar o classificador mais apropriado para cada instância do problema.

2.2.1 Aprendizagem-base, meta-aprendizagem e meta-conhecimento

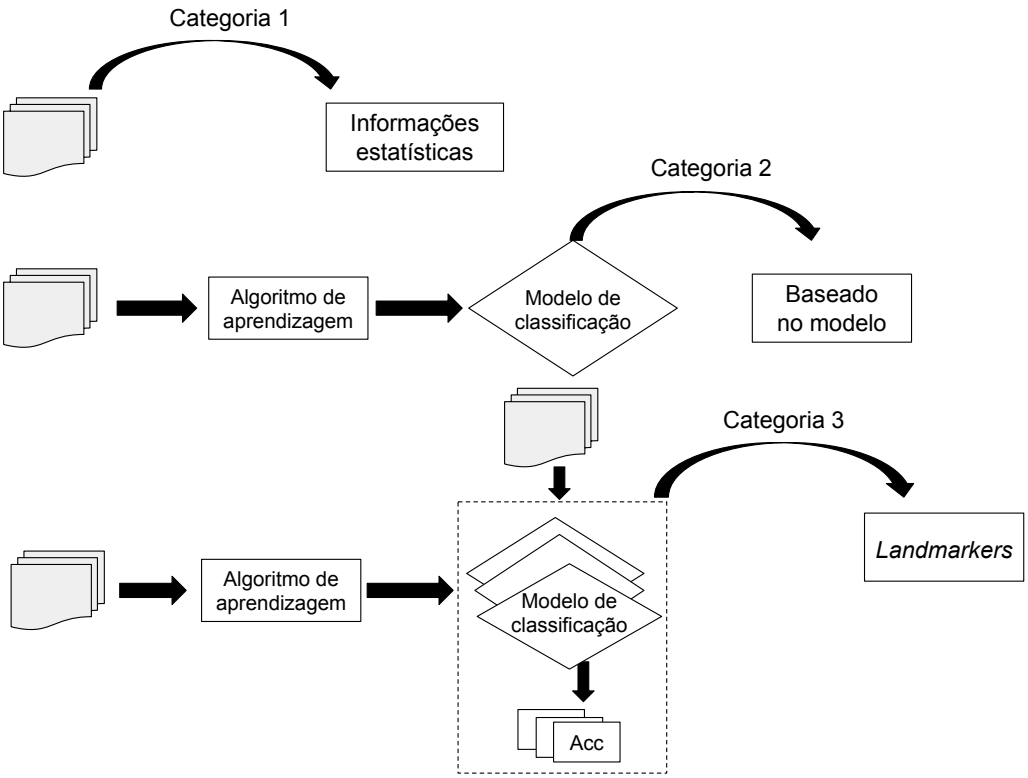
No processo de aprendizagem tradicional, ou aprendizagem-base, normalmente, quanto maior o número de instâncias passados ao algoritmo de aprendizagem, maior a probabilidade do modelo de aprendizagem criado ser capaz de conseguir classificar novos exemplos com eficácia. Porém, esse tipo de aprendizagem é limitado, pois não explora os padrões encontrados nos dados por meio do treinamento do modelo para seu próprio aprendizado, ou seja, não é adquirida nenhuma forma de experiência com os consecutivos experimentos realizados com um mesmo conjunto de dados. Nesse contexto também, a transferência de conhecimento entre domínios é inviável. A meta-aprendizagem, por sua vez, explora o conhecimento adquirido por meio do processo de aprendizagem tradicional e aprende com as características dele. Ela está presente em adaptações realizadas no processo de aprendizagem tradicional, como adicionar, alterar ou mesmo combinar técnicas de aprendizagem (BRAZDIL et al., 2009).

O conhecimento adquirido por meio do processo de aprendizagem-base é definido na literatura como meta-conhecimento. Segundo Brazdil et al. (2009), esse tipo de conhecimento pode conter informações como algoritmos que apresentaram uma boa performance, métricas de similaridade entre conjunto de dados, assim como qualquer informação proveniente de um processo de aprendizagem. Um objetivo importante da meta-aprendizagem é estudar formas de extrair e explorar o meta-conhecimento.

2.2.2 Meta-atributos

Os meta-atributos são uma forma de meta-conhecimento gerados a partir da fase inicial do processo de aprendizagem. Dentre as categorias de meta-atributos, podem ser citadas: caracterização estatística e informação teórica, baseados no modelo e baseados no desempenho de um conjunto de algoritmos de aprendizagem (BRAZDIL et al., 2009). A Figura 4.1 demonstra as categorias supracitadas.

Figura 2.15 – Categorias de meta-atributos.



Fonte: Adaptada de Brazdil et al. (2009).

A primeira categoria de meta-atributos refere-se às características do conjunto de dados. O objetivo é utilizar características descritivas dos conjuntos de dados e tentar identificar alguma correlação com o desempenho dos algoritmos de aprendizagem. Exemplos de meta-atributos dessa categoria são: número de classes, número de atributos, a fração de exemplos por atributos etc.

Os meta-atributos da segunda categoria são baseados no modelo de aprendizagem. Nesse caso, eles podem ser as informações obtidas a partir do processo de geração do modelo. Como exemplos desses atributos podem ser citadas as propriedades de uma árvore de decisão construída a partir de um conjunto de dados, tais como o formato da árvore, a sua profundidade máxima, a quantidade de nós por atributo, desbalanceamento da árvore etc.

Já na terceira categoria, busca-se explorar informação obtida sobre o conjunto de algoritmos de aprendizagem que obtiveram desempenhos diferentes para um determinado conjunto de dados e espera-se que os meta-atributos identificados possam ser úteis para encontrar aquele algoritmo de aprendizagem que vai ser especialista em determinada tarefa. Esse conjunto de algoritmos de aprendizagem é denominado pontos de referência, do inglês *landmarkers* (BRAZDIL et al., 2009). Como exemplo de atributo dessa categoria, pode ser citada a acurácia obtida

no desempenho dos algoritmos. Por meio do processo de *landmarking*, a acurácia é utilizada como critério de otimização (BALTE et al., 2014).

Este capítulo, iniciado pela grande área de Mineração de Dados, o KDD e suas etapas, apresentou a fundamentação teórica necessária para a compreensão deste trabalho. A Mineração de Texto foi introduzida juntamente com o Processamento de Linguagem Natural e suas técnicas. Como a tarefa de mineração de dados aplicada ao trabalho é a classificação, foram apresentadas algumas técnicas para a sua realização. Por fim, os tópicos principais foram abordados, os quais são: Classificação Multirrótulo com alguns de seus métodos e métricas de avaliação e Meta-aprendizagem. No capítulo seguinte são apresentados alguns trabalhos relacionados à este que já aplicaram a meta-aprendizagem em problemas de classificação.

3 TRABALHOS RELACIONADOS

Este capítulo descreve alguns trabalhos apresentados na literatura que estão relacionados com o tema desta pesquisa. Alguns desses trabalhos focaram no desenvolvimento de meta-classificadores para a classificação monorrótulo, multirrótulo e para a classificação hierárquica. Já outros mostraram a aplicação do aprendizado de máquina automatizado, do inglês *Automated Machine Learning* (AutoML), para a classificação multirrótulo. A abordagem AutoML já é muito utilizada na classificação monorrótulo e uma de suas denominações na literatura é meta-aprendizagem construtiva (SÁ et al., 2017). Neste trabalho, a meta-aprendizagem é aplicada na classificação multirrótulo especificamente para problemas de domínio textual.

A Tabela 3.1 apresenta as principais características dos trabalhos encontrados na literatura e a correlação entre os mesmos e o método proposto neste trabalho. Nessa tabela, a primeira coluna refere-se aos métodos apresentados neste capítulo, a segunda refere-se ao tipo de classificação abordada nos trabalhos, a terceira mostra se a metodologia adotada foi empregada para seleção de classificadores por conjunto de dados ou por instância, a quarta apresenta o método de indução utilizado na fase de classificação de novos dados e a última coluna refere-se ao domínio específico do trabalho.

Tabela 3.1 – Correlação entre os trabalhos relacionados

Método	Tipo de classificação	Tipo de seleção	Método de indução	Domínio
(ZHANG et al., 2017)	monorrótulo	instância	similaridade	variado
(TRIPATHI et al., 2015)	hierárquica	conjunto de dados	baseado em busca	texto
(CHEKINA et al., 2011)	multirrótulo	conjunto de dados	similaridade	variado
(SÁ et al., 2017)	multirrótulo	conjunto de dados	algoritmo evolutivo	variado
(SÁ et al., 2018)	multirrótulo	conjunto de dados	programação genética	variado
Método proposto	multirrótulo	instância	similaridade	texto

Fonte: Elaborada pela autora (2020).

Em (ZHANG et al., 2017) foi proposto um modelo de seleção de classificadores que aplica a meta-aprendizagem no contexto monorrótulo. Denominado *Discriminative Model Selection* (DMS), esse modelo escolhe modelos de classificação únicos e potencialmente diferentes para cada instância a ser classificada. Essa solução foi idealizada em consequência da difi-

culdade de se encontrar um classificador apropriado para um determinado conjunto de dados e das desvantagens dos modelos de classificação híbridos. A abordagem do DMS é composta por duas fases: treinamento e classificação. Na fase de treinamento, entre dois modelos, encontra-se o melhor para cada instância da base de dados de treinamento. A partir dessa informação, na fase de classificação é aplicada uma medida de similaridade para encontrar a instância de treinamento mais próxima da instância que se deseja classificar. Em seguida, o DMS escolhe o classificador associado à instância de treinamento mais similar para classificar a nova instância. Foram realizados experimentos para validar a eficácia do DMS para a classificação de texto. Para isso, foram utilizados 19 conjuntos de dados textuais. Os classificadores base utilizados no DMS para a classificação de texto foram *Multinomial Naive Bayes* (MNB) e *Complement Naive Bayes* (CNB). Os resultados mostraram que o DMS conseguiu superar o desempenho dos classificadores base com uma acurácia média de 85,17% contra uma acurácia de 82, 44% do classificador MNB e 84,12% do classificador CNB.

O método proposto por Zhang et al. (2017), embora não seja aplicado à classificação multirrótulo, é o que mais apresenta características similares ao método proposto neste trabalho. Os autores aplicaram a meta-aprendizagem para problemas monorrótulo, direcionados à conjunto de dados de tipos variados, e utilizaram apenas dois classificadores base no modelo desenvolvido. Na aplicação do DMS para textos, na fase de classificação, utilizou-se a similaridade do cosseno e baseou-se em apenas uma instância mais similar. Além disso, a técnica empregada no treinamento dos modelos difere-se da aplicada neste trabalho quanto à forma de divisão do conjunto de dados; assim como na avaliação para escolha dos modelos.

Tripathi et al. (2015) propuseram um meta-classificador escalável para abordar o problema de classificação hierárquica de textos a partir de grandes conjuntos de dados. Os autores utilizaram uma técnica para reduzir a dimensão da representação dos dados. Essa técnica é denominada aprendizagem subespacial. O método proposto por eles tem uma estrutura para classificação de texto de dois níveis e utiliza uma representação vetorial especial chamada vetor de significância condicional que representa uma hierarquia de categoria de dois níveis com um único vetor e incorpora conteúdo semântico nos vetores. Esse vetor é composto pelas categorias (classes) de nível-1 dos documentos e suas subcategorias, as quais são posicionadas consecutivamente no vetor. Dessa forma, o espaço vetorial é dividido em subespaços que representam cada categoria principal (nível-1) do documento. O vetor de significância de cada documento contém um valor que representa a sua associação com cada categoria e o valor máximo prova-

velmente é a categoria real do documento. No treinamento, utiliza-se um classificador para cada subespaço presente, ou seja, diferentes instâncias de um classificador são treinadas para cada categoria principal. Para agilizar o processo de classificação, eles utilizaram um método baseado em busca para detectar a categoria de nível-1 do documento a ser classificado e o classificador treinado neste subespaço é ativado para classificação de nível-2 do documento. Para testar a arquitetura do meta-classificador, foram utilizados os seguintes classificadores base: *random forest*, C4.5, *multilayer perceptron*, *naive Bayes*, *BayesNet*(BN) e PART. Foram utilizados dois conjuntos de dados textuais nos experimentos e, de acordo com os resultados, apesar de haver pouca variação no desempenho do meta-classificador com as diferentes arquiteturas, todos eles apresentaram um desempenho muito melhor do que os classificadores individuais. Foi constatado pelos autores que o bom desempenho da arquitetura construída é devido ao método de combinação dos classificadores e não nos classificadores individualmente.

O meta-classificador apresentado por Tripathi et al. (2015) aplicou a meta-aprendizagem na classificação hierárquica de textos. Diferentemente da classificação multirrótulo, na classificação hierárquica as classes encontram-se organizadas em diferentes níveis de uma hierarquia. Assim como neste trabalho, Tripathi et al. (2015) aplicaram a meta-aprendizagem para encontrar o classificador mais apropriado para a classificação de textos. No entanto, o método proposto por eles objetivou encontrar o melhor classificador para cada classe apresentada no problema de classificação, enquanto neste trabalho, o foco é selecionar o classificador mais apropriado para cada instância a ser classificada baseado na similaridade entre essas e as instâncias do conjunto de treinamento.

Em (CHEKINA et al., 2011), os autores apontam a escassez de informações relacionadas a eficácia dos algoritmos de classificação multirrótulo e apresentam uma abordagem que aplica a meta-aprendizagem para selecionar o algoritmo de classificação multirrótulo mais eficaz para um determinado conjunto de dados. Essa seleção é baseada na descrição das características dos conjuntos de dados, que são utilizadas para criar regras ou ferramentas, as quais ajudam a identificar o algoritmo ótimo para um determinado problema. Para alcançar os objetivos desejados, os autores avaliaram vários algoritmos de classificação multirrótulo utilizando um repositório de conjuntos de dados artificialmente extendido. O objetivo da avaliação era comparar o desempenho dos diferentes classificadores e desenvolver um meta-classificador para predizer o classificador mais apropriado para um determinado conjunto de dados. Os resultados dos experimentos demonstraram a eficiência da meta-aprendizagem aplicada. Os autores assumem que

se um algoritmo supera o desempenho dos outros na classificação de um conjunto de dados, ele, consequentemente, é o mais apropriado para outros conjuntos de dados que apresentam características semelhantes.

Chekina et al. (2011) aplicaram a meta-aprendizagem na classificação multirrótulo de conjunto de dados genéricos. Os autores extraíram uma série de meta-atributos dos conjuntos de dados e os utilizaram no processo de aprendizagem. No método apresentado por eles, as características descritivas dos conjuntos de dados desempenham um papel importante na classificação de novos dados. Embora o meta-classificador proposto em (CHEKINA et al., 2011) seja aplicado à classificação multirrótulo, ele não tem o foco específico para textos e não seleciona os classificadores individualmente para cada instância do conjunto de dados, cujos aspectos são endereçados neste trabalho.

Similarmente à aplicação da meta-aprendizagem, alguns trabalhos, como (SÁ et al., 2017) e (SÁ et al., 2018), têm utilizado o aprendizado de máquina automatizado com o mesmo objetivo de selecionar o melhor classificador para um determinado conjunto de dados multirrótulo. O primeiro método proposto com a aplicação do aprendizado de máquina automatizado para o cenário multirrótulo foi apresentado por Sá et al. (2017). Os autores desenvolveram o primeiro algoritmo evolutivo para, de forma automática, selecionar o melhor algoritmo de classificação multirrótulo com sua melhor configuração para um conjunto de dados multirrótulo. O espaço de busca compreendia 31 algoritmos de classificação multirrótulo com seus parâmetros. O método foi avaliado utilizando três conjuntos de dados multirrótulo e seu desempenho foi comparado com os métodos relevância binária e cadeia de classificadores. Os resultados mostraram a competitividade do método em relação aos outros comparados. Já em (SÁ et al., 2018), os autores apresentaram um método para a seleção e configuração de algoritmos de classificação multirrótulo com programação genética baseada em gramática. Esse método, denominado Auto-MEKA$_{GGP}$, seleciona, dentre vários algoritmos para a classificação multirrótulo e suas respectivas configurações, presentes na ferramenta MEKA, aquele que apresenta o melhor desempenho para um conjunto de dados fornecido como entrada. As possibilidades de algoritmos multirrótulo disponíveis, aliadas às suas configurações, são representadas por uma gramática, que é utilizada pelo método de programação genética para buscar o melhor classificador. Para a avaliação do método foram utilizados 10 conjuntos de dados multirrótulo e seu desempenho foi comparado com outros quatro métodos para a classificação multirrótulo. Dentre esses, um método proposto anteriormente pelos mesmos autores, Relevância Binária, Cadeia de Classifi-

cadores e uma versão simplificada do próprio Auto-MEKA$_{GGP}$. Os resultados obtidos mostraram a eficiência do método sobre os demais, ao apresentar desempenho superior em 60% dos conjuntos de dados utilizados.

Os trabalhos apresentados em (SÁ et al., 2017) e (SÁ et al., 2018) utilizaram AutoML no processo de seleção do algoritmo de classificação mais apropriado para um determinado problema multirrótulo. Assim como na aplicação da meta-aprendizagem abordada neste trabalho, o objetivo dos autores é automatizar o processo de seleção de classificadores. Porém, a AutoML, aplicada no processo de aprendizagem dos métodos apresentados por eles, não realiza a seleção dos classificadores mais apropriados por instância e nem é específico para texto.

Nos trabalhos supramencionados têm-se a aplicação de meta-aprendizagem para bases de dados textuais no contexto monorrótulo, meta-aprendizagem para bases de dados não textuais no contexto multirrótulo e métodos que utilizam AutoML para a classificação multirrótulo. A abordagem desenvolvida neste trabalho, por meio da meta-aprendizagem, visa selecionar o(s) classificador(es) mais apropriado(s) para cada instância de texto que se deseja classificar.

O objetivo deste capítulo foi apresentar abordagens semelhantes à desenvolvida nesta dissertação. Para isso, foram selecionados aqueles trabalhos que utilizaram a meta-aprendizagem como foco. O próximo capítulo descreve o método proposto e apresenta a especificação do seu algoritmo.

4 MÉTODO PROPOSTO

Este capítulo apresenta a descrição do método proposto com os passos adotados para a sua implementação. Como apresentado no Capítulo 1, alguns trabalhos recentes têm aplicado a meta-aprendizagem no processo de seleção do classificador mais apropriado para um determinado conjunto de dados. No entanto, não foram encontrados trabalhos que aplicaram a meta-aprendizagem especificamente para classificação de texto no contexto multirrótulo e de forma individualizada, ou seja, para cada instância a ser classificada. Portanto, o objeto principal desta investigação é a criação de um método que utiliza a meta-aprendizagem para a escolha de um classificador multirrótulo adequado para cada instância de texto de uma coleção. A Seção 4.1 apresenta uma descrição geral da abordagem proposta. Logo em seguida, na Seção 4.2, são apresentados os detalhes do método proposto.

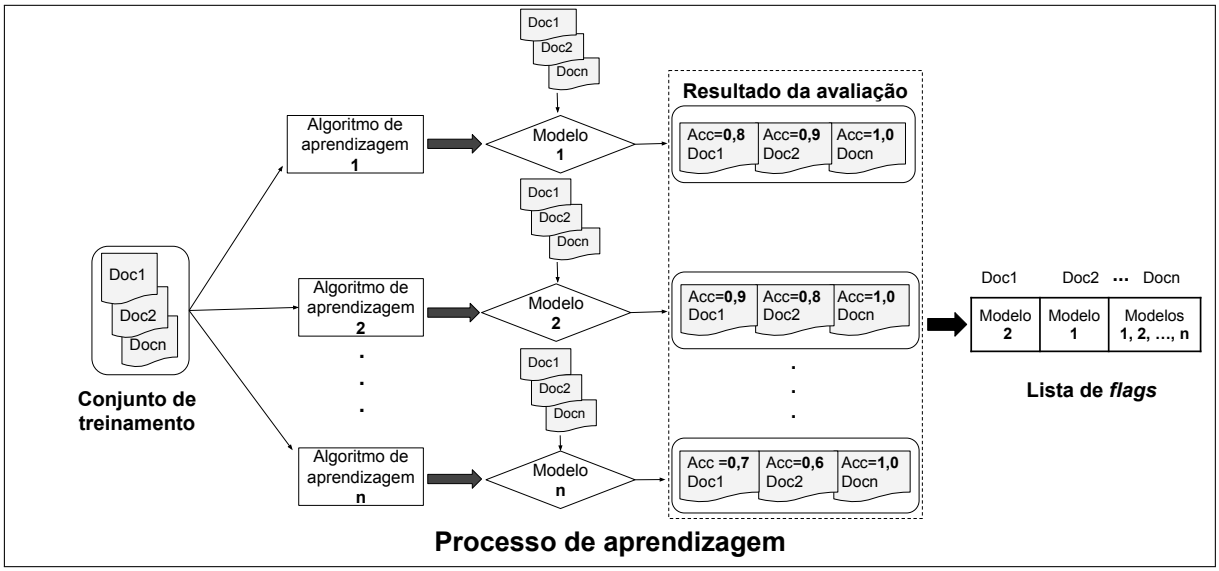
4.1 Descrição do método proposto

O método proposto para classificação de texto multirrótulo foi desenvolvido com o objetivo de automatizar a difícil tarefa de escolher algoritmo(s) de classificação para um determinado conjunto de dados. No caso da abordagem aqui proposta, a escolha do classificador é realizada para cada instância, uma vez que diferentes classificadores podem ter desempenhos distintos para diferentes instâncias. O método proposto é denominado MetaMLC (*Meta for Multilabel Classification*) e seu processamento pode ser dividido em duas fases: treinamento e classificação.

Na fase de treinamento, representada na Figura 4.1, o método avalia o desempenho de um conjunto de classificadores multirrótulo. A avaliação é realizada utilizando a métrica *Hamming Loss* (ver Seção 2.1.11), muito utilizada na avaliação de classificadores multirrótulo. Essa métrica permite contabilizar o erro na predição para cada instância, de forma individual. Nessa etapa, inicialmente, os classificadores são avaliados globalmente (considerando todas as instâncias do conjunto de treinamento) para encontrar aquele que tem o melhor desempenho geral. Essa informação do desempenho geral dos classificadores será utilizada se houver empate entre os classificadores na fase de treinamento. Além disso, os classificadores também são avaliados localmente, ou seja, para cada instância presente no conjunto de treinamento. Dessa forma, constrói-se uma lista de *flags* que armazena, para cada instância de treino, o identificador do(s) classificador(es) que apresenta(m) o melhor desempenho.

O método proposto, MetaMLC, é composto por três classificadores base. Na avaliação de desempenho de cada instância individualmente, um empate nessa avaliação entre dois classificadores é solucionado com aquele que obteve o maior desempenho global. Se na avaliação de desempenho ocorrer empate entre os três classificadores base, o identificador armazenado na lista de *flags* refletirá os três, o que resultará em um comitê para a fase de classificação.

Figura 4.1 – Fase 1 - Treinamento.

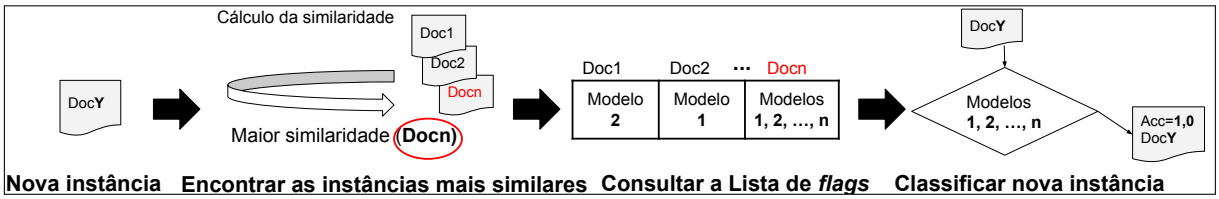


Fonte: Elaborada pela autora (2020).

Já na fase de classificação, representada na Figura 4.2, dada uma instância para ser classificada, buscam-se as três instâncias mais semelhantes a ela no conjunto de treinamento. De posse das instâncias mais similares e da lista de *flags* construída na etapa de treinamento, define-se qual será o conjunto de modelos utilizado para classificar essa nova instância.

Na fase de classificação, optou-se pela escolha de três instâncias mais similares para aumentar a certeza na identificação do classificador mais apropriado. Como foram utilizados três classificadores base, a predominância de um deles na vizinhança (posições na lista de *flags* referentes às três instâncias mais similares) o apontaria como o melhor para classificar a nova instância. Caso contrário, o comitê formado pelos três é utilizado.

Figura 4.2 – Fase 2 - Classificação.



Fonte: Elaborada pela autora (2020).

4.2 Especificação dos algoritmos

Os pseudocódigos dos algoritmos responsáveis pelas etapas de treinamento e classificação são apresentados em Algoritmo 1 e Algoritmo 2, respectivamente.

Algoritmo 1 *Fase de treinamento*

Entrada: conjunto de dados de treinamento X, um conjunto de algoritmos de classificação $C = \{C_1, C_2, ..., C_l\}$.
Saída: modelos treinados $\{M_1, M_2, ..., M_l\}$ e uma lista de *flags* F.

1: **Início**
2: A partir de X, os modelos M_1, M_2, ..., M_l são treinados utilizando-se os classificadores C_1, C_2, ..., C_l, respectivamente.
3: O conjunto X é particionado em treino e teste, modelos m_1, m_2, ..., m_l são treinados utilizando-se os classificadores C_1, C_2, ..., C_l, respectivamente, com o objetivo de encontrar o melhor desempenho global.
4: **para cada** (instância $x_i \in X$) **faça**
5: $\quad M^{'} \leftarrow \emptyset$
6: $\quad$ **para cada** $j = 1$ a l **faça**
7: $\quad\quad M^{'}_j \leftarrow$ modelo treinado utilizando C_j a partir do conjunto de treinamento $T = \{X - x_i\}$
8: $\quad\quad M^{'} \leftarrow M^{'} \cup M^{'}_j$
9: $\quad$ **fim para cada**
10: $\quad B_i \leftarrow$ melhores modelos avaliados $\in M^{'}$ para a instância x_i
11: $\quad F[x_i] \leftarrow$ identificadores dos modelos $\in B_i$
12: **fim para cada**
13: **retorna** $M = \{M_1, M_2, ..., M_l\}$ e a lista de *flags* F.
14: **Fim**

Para a fase de treinamento, o Algoritmo 1 recebe como entrada um conjunto de treinamento $X = \{x_1, x_2, ..., x_n\}$, em que cada instância x_i está associada a um conjunto de rótulos Y_i, e um conjunto de algoritmos de classificação multirrótulo $C = \{C_1, C_2, ..., C_l\}$. Como saída, o algoritmo retorna um conjunto de modelos treinados $M = \{M_1, M_2, ..., M_l\}$ e uma lista de *flags*.

Inicialmente, na linha 2, os modelos de classificação, que serão retornados, são criados utilizando cada algoritmo do conjunto C a partir das instâncias do conjunto de dados X. Em seguida, usando o método de validação cruzada com k partições, o conjunto de dados de treinamento X é usado para avaliar globalmente cada algoritmo de classificação em C (linha 3). O objetivo dessa avaliação é encontrar o modelo que alcança o melhor desempenho geral para ser usado como desempatador na avaliação individual de cada instância, que é realizada nos próximos passos.

A partir da linha 4, inicia-se o processo para encontrar o conjunto de modelos mais apropriado para cada instância do conjunto X. Essa busca é realizada com o emprego da técnica de validação cruzada *leave-one-out*, ou seja, os modelos de classificação são treinados a partir

de $T = \{X - x_i\}$ (linhas 6 à 9) e avaliados para cada instância x_i. Com os modelos criados, o próximo passo é encontrar os que são mais precisos para cada instância de X. Para encontrar esses modelos, para cada instância x_i, usamos a métrica *Hamming Loss* para comparar o desempenho preditivo dos modelos treinados a partir do conjunto de treinamento T (linha 10). A partir dos resultados das comparações, é criada uma lista de *flags* F, contendo em cada posição o(s) identificador(es) do(s) modelo(s) mais preciso(s) para cada instância investigada (linha 11).

Na avaliação dos resultados de desempenho dos modelos (linha 10), se houver a ocorrência de empate entre dois modelos para uma determinada instância, o modelo que obteve desempenho global superior (definido na linha 3) será escolhido como o mais apropriado. Se houver empate entre mais de dois modelos, será formado um comitê incluindo todos eles. Nesse caso, o valor a ser preenchido na lista de *flags* sinalizará a ocorrência de todos os modelos. Caso contrário, será selecionado, individualmente, o modelo que teve o melhor desempenho para a instância avaliada. Com a lista de *flags* preenchida, termina a fase de treinamento e o algoritmo a retorna juntamente com o conjunto de modelos construídos M (linha 13).

Algoritmo 2 *Fase de classificação*

Entrada: o conjunto de treinamento X, uma instância a ser classificada y, o conjunto de modelos treinados M e a lista de *flags* F.

Saída: Y, um conjunto de rótulos preditos para y.

1: **Início**
2: $S \leftarrow$ conjunto contendo as três instâncias $\in X$ mais similares à y.
3: $C \leftarrow \emptyset$
4: **para cada** $s \in S$ **faça**
5: $\quad C \leftarrow C \cup F[s]$
6: **fim para cada**
7: **se** $|C| = 1$ **então**
8: $\quad Y \leftarrow$ classifica y utilizando o modelo treinado M_j, onde $j \subset C$
9: **senão**
10: $\quad Y \leftarrow$ classifica y utilizando um comitê composto por todos os modelos treinados $\in M$
11: **fim se**
12: **retorna** Y
13: **Fim**

Na fase de classificação, o primeiro passo é encontrar as instâncias do conjunto de treinamento X que são mais similares à instância que se deseja classificar. Para encontrar a similaridade entre as instâncias, o método proposto utiliza a similaridade do cosseno (ver Seção 2.1.6).

O Algoritmo 2 descreve o processo de classificação. Ele recebe como entrada o conjunto de modelos treinados M e a lista de *flags* F obtidos na fase de treinamento, o conjunto de

treinamento X, e uma instância y para ser classificada. Como saída, o algoritmo retorna o conjunto de rótulos preditos para y.

A partir da linha 2, o conjunto S com os identificadores das três instâncias no conjunto de treinamento X mais semelhante a y é criado. Em seguida, acessando a lista de *flags* F, é criado o conjunto C contendo os identificadores dos modelos de classificação mais apropriados para as instâncias em S (linhas 3-6). Logo após, verifica-se se C é um conjunto unitário, ou seja, se existe um único classificador associado às instâncias em S (linha 7). Se verdadeiro, o modelo de classificação em C é usado para classificar a nova instância; caso contrário, um comitê formado por todos os classificadores do conjunto M é usado (linha 10). Quando um comitê de classificadores é usado, o resultado da predição é dado pela regra da votação majoritária, conforme exemplificado na Tabela 4.1, que considera três modelos de classificação. Como resultado dessa etapa, o algoritmo retorna o conjunto de rótulos preditos para a instância y (linha 12).

Na Tabela 4.1, a primeira coluna apresenta os modelos de classificação e as colunas seguintes possuem os oito rótulos possíveis para uma determinada instância y. Cada linha da tabela exibe a predição do modelo para cada rótulo, ou seja, o valor 1 se o rótulo dessa coluna foi predito e o valor 0 caso contrário. Considerando cada rótulo individualmente, a predição final para y (última linha) é o resultado da regra da votação majoritária. Neste exemplo, o conjunto de rótulos preditos para a instância y é $Y = \{r1, r5, r6, r7, r8\}$

Tabela 4.1 – Exemplo da combinação dos resultados dos classificadores para uma dada instância y

Modelos	r1	r2	r3	r4	r5	r6	r7	r8
M_1	1	0	0	1	1	1	1	1
M_2	0	1	1	0	0	0	0	1
M_3	1	0	0	0	1	1	1	0
Comitê (M1+M2+M3):	1	0	0	0	1	1	1	1

Fonte: Elaborada pela autora (2020).

Vale ressaltar que no desenvolvimento do método MetaMLC foram utilizados três classificadores base. Dessa forma, as estratégias de desempate empregadas estão intrinsicamente alinhadas ao número de classificadores utilizados.

Este capítulo apresentou a descrição do método proposto neste trabalho e a especificação dos seus algoritmos. O próximo capítulo apresenta a análise experimental realizada com os detalhes da configuração experimental, avaliação dos resultados e discussão.

5 ANÁLISE EXPERIMENTAL

Este capítulo detalha os passos adotados na realização dos experimentos computacionais, assim como apresenta as ferramentas utilizadas na implementação do método proposto e no processo de classificação como um todo. A Seção 5.1 apresenta a configuração experimental com todas as ferramentas e técnicas utilizadas. Logo em seguida, na Seção 5.2, são reportados os resultados experimentais e a discussão.

5.1 Configuração experimental

O método proposto foi implementado na linguagem Python e as principais ferramentas utilizadas foram *Scikit Learn* (PEDREGOSA et al., 2011), biblioteca para mineração de dados e aprendizagem de máquina, e *Scikit Multilearn* (SZYMAŃSKI; KAJDANOWICZ, 2019), uma biblioteca específica para trabalhar com problemas de classificação multirrótulo. Na sequência, a Seção 5.1.1 descreve as bases de dados utilizadas; Seção 5.1.2 apresenta o método utilizado no particionamento do conjunto de dados e suas características; Seção 5.1.3 apresenta as técnicas utilizadas na etapa de pré-processamento e representação dos documentos; Seção 5.1.4 reporta a configuração dos classificadores utilizados como base para o método proposto e, por fim, a Seção 5.1.5 aponta a métrica de similaridade utilizada.

5.1.1 Bases de dados utilizadas

Para avaliar o método proposto foram realizados experimentos com a utilização de duas bases de dados com textos escritos em português, denominadas BFRC-PT e G1, disponibilizadas por Curi et al. (2018) e quatro bases de dados com textos escritos em inglês, denominadas Enron (documentos de e-mail), Medical (prognóstico médico), Ohsumed (literatura médica) e Slashdot (títulos e resumos de artigos de tecnologia). Essas bases de dados foram adquiridas por meio do site da ferramenta MEKA [1].

A base de dados BFRC-PT, criada pelos pesquisadores Curi et al. (2018), trata-se de um conjunto de 8.080 documentos de notícias de entretenimento coletadas da versão brasileira do *BuzzFeed* [2].

[1] https://sourceforge.net/projects/meka/files/Datasets/Prefiltered/
[2] https://www.buzzfeed.com/br

Já a base de dados G1 possui um total de 2.000 instâncias que correspondem a títulos de notícias extraídos do *website* G1 [3]. Ambos os conjuntos de dados escritos em português são multirrótulo e foram criados com intuito de classificar reações de usuários ao lerem notícias *online*.

O conjunto de dados Enron é um subconjunto de documentos de texto multirrótulo proveniente de um total aproximado de 500 mil mensagens de e-mail. Esse subconjunto de dados foi rotulado por um grupo de estudantes do curso de Processamento de Linguagem Natural Aplicada da University of California em Berkeley. A seleção dos e-mails foi focada em assuntos relacionados à crise de energia da cidade de Califórnia e à negócios e, com isso, evitou-se e-mails de cunho pessoal. No total, foram escolhidos 53 rótulos [4] para a classificação dos documentos de e-mail.

Medical é um conjunto de dados de texto multirrótulo referentes à prognóstico médico apresentado em (PESTIAN et al., 2007). Esse conjunto de dados surgiu a partir do *The 2007 Computational Medicine Challenge* com o objetivo de criar um conjunto de dados do domínio médico que fosse cuidadosamente anonimizado e pudesse ser distribuído livremente e também pudesse contribuir com o avanço na área de mineração de dados clínicos. Conforme Pestian et al. (2007), textos clínicos livres apresentam um desafio maior para as ferramentas de processamento de linguagem natural devido a forma técnica empregada na escrita dos médicos, como jargões, abreviações etc. Contudo, os autores enfatizam a importância da classificação desses dados para a área de saúde e, consequentemente, economia. Os documentos de texto foram coletados do Departamento de Radiologia do *Cincinnati Children's Hospital Medical Center*. Os documentos de texto coletados são representativos da atividade pediátrica em radiologia e são referentes a relatórios de radiologia de ex-pacientes de um período de um ano. O conjunto de dados Medical utilizado neste trabalho possui um total de 45 rótulos, os quais são códigos de diagnóstico médico do sistema Internacional de Classificação de Doenças, Nona Revisão, Modificação Clínica, do inglês *International Classification of Diseases, Ninth Revision, Clinical Modification* (ICD-9-CM). Os documentos foram rotulados pelos funcionários do hospital supramencionado e por duas empresas independentes.

O conjunto de dados Ohsumed é composto por documentos de textos da literatura médica referentes ao grupo de doenças cardiovasculares. Para a sua criação, foram selecionados 20.000 documentos de texto de um total de 50.216 resumos médicos do ano de 1991. Desses

[3] https://g1.globo.com/

[4] http://bailando.sims.berkeley.edu/enron_email.html

20.000, 13.929, relacionados à doenças cardiovasculares, compuseram o conjunto Ohsumed. Os documentos de texto são rotulados pelo assunto médico dos títulos dos trabalhos e totalizam 23 categorias.

Já o conjunto de dados Slashdot é composto por um total de 3782 instâncias de texto referentes a títulos de notícias e sinopses parciais relacionadas a tecnologia. Os textos para a formação desse conjunto de dados foram extraídos do *web site* Slashdot [5]. O conjunto possui 22 rótulos e chega à um total de 156 combinações únicas de rótulos.

As características dos conjuntos de dados, a saber, número de instâncias (# de instâncias), número de rótulos (|L|), número médio de rótulos por instância, (Cardinalidade), a normalização da cardinalidade, (Densidade), número de conjuntos de rótulos distintos (|LS|), número médio de instâncias por rótulo e o seu desvio padrão (Balanceamento) e o idioma (Pt = português e En = inglês) são apresentadas na Tabela 5.1. Tais características demonstram a diversidade dos conjuntos de dados utilizados.

Tabela 5.1 – Características dos conjuntos de dados utilizados

Nome	**# de instâncias**	**|L|**	**Cardinalidade**	**Densidade**	**|LS|**	**Balanceamento**	**Idioma**
G1	2000	7	1,964	0,280	70	561,00±210,62	Pt
BRFC-PT	8080	8	3,861	0,483	206	3899,25±1616,73	Pt
Enron	1702	53	3,378	0,064	753	108,49±197,57	En
Medical	978	45	1,245	0,028	94	27,06±47,95	En
Ohsumed	13929	23	1,663	0,072	1147	1007,21±905,87	En
Slashdot	3782	22	1,18	0,05	156	203,00±190,84	En

Fonte: Elaborada pela autora (2020).

5.1.2 Divisão do conjunto de dados e avaliação

A divisão dos conjuntos de dados, em treino e teste, foi realizada com a validação cruzada de três partes com o método *IterativeStratification* (SECHIDIS et al., 2011; SZYMAŃSKI; KAJDANOWICZ, 2017). Basicamente, esse método divide um conjunto de dados multirrótulo em vários subconjuntos disjuntos de tamanhos aproximadamente igual, tentando manter a pro-

[5] https://slashdot.org/

porção de instâncias de cada rótulo de classe em cada subconjunto aproximadamente igual à do conjunto de dados completo.

Na estratificação dos conjuntos de dados multirrótulo, duas vertentes são consideradas: 1) partição por combinações dos rótulos e 2) partição por exemplos positivos e negativos de cada rótulo. A abordagem considerada no método utilizado considera a segunda vertente. Dessa forma, o processo de divisão iterativo se inicia pelos rótulos que apresentam um número menor de instâncias e segue na ordem crescente.

A quantidade de partições para a validação cruzada foi definida de acordo com o trabalho realizado por Curi et al. (2018), que levou em conta o desbalanceamento dos dados para definir a quantidade de partições. Como observado na Tabela 5.1, o desbalanceamento também é uma característica dos conjuntos de dados utilizados nos experimentos deste trabalho.

5.1.3 Pré-processamento e representação dos documentos

Essa etapa inicial do processo de mineração de textos foi executada com a utilização do kit de ferramentas para processamento de linguagem natural (NLTK) e, após processados, os textos foram vetorizados por meio da técnica *word embeddings* Word2Vec (MIKOLOV et al., 2013a). As técnicas de PLN aplicadas nos textos foram as seguintes:

- tokenização: com a utilização da ferramenta *RegexpTokenizer*, a qual utiliza uma expressão regular para dividir um documento de textos em *tokens*.
- conversão para letras minúsculas: foi utilizado o método *lower*() para converter todas as palavras maiúsculas para a forma minúscula.
- remoção de *stop words*: realizada por meio da ferramenta *stopwords*, a qual possui listas de palavras (*stop words*) para várias linguagens.

Após aplicadas as técnicas supracitadas, para os conjuntos de dados escritos em português foi utilizado o modelo Word2Vec pré-treinado, com 300 dimensões, criado pelo Núcleo Interinstitucional de Linguística Computacional (NILC) (HARTMANN et al., 2017). Esse Word2Vec foi treinado com uma extensa coleção composta por diferentes gêneros textuais escritos em português do Brasil e europeu. Para o treinamento do recurso foram utilizados 17 córpus, que totalizaram 1,395,926,282 de *tokens*. Dentre os gêneros textuais presentes na coleção estão o informativo, didático, prosa e comunicação científica.

Já para os conjuntos de dados escritos em inglês foi utilizado o Word2Vec disponibilizado pelo Google [6], com 300 dimensões para três milhões de palavras e frases. O Word2Vec foi treinado com um conjunto de dados sobre notícias do Google com mais de um bilhão de palavras. Essa ferramenta é descrita em (MIKOLOV et al., 2013b).

Para a transformação dos documentos de texto utilizando o modelo Word2Vec pré-treinado, foi utilizada a média dos vetores de cada palavra.

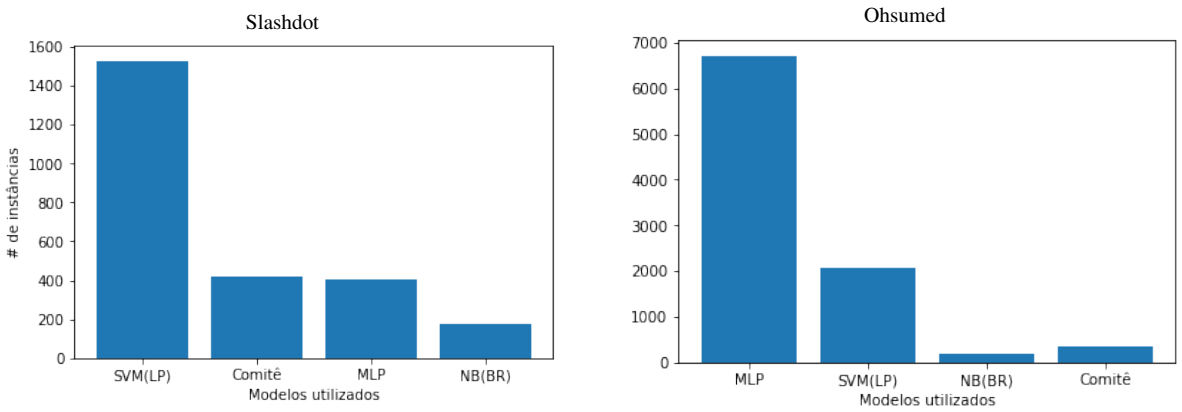
5.1.4 Configuração dos métodos de classificação

Foram realizados vários experimentos com os métodos de classificação multirrótulo mais utilizados na literatura e diferentes tipos de pré-processamento e representação dos documentos, como os apresentados no Capítulo 2, com o intuito de identificar aqueles que seriam utilizados como classificadores base para o método proposto. Com os resultados obtidos a partir da avaliação dos classificadores nos conjuntos de dados utilizados, foi selecionado um conjunto de três classificadores para compor o método proposto. Os métodos escolhidos foram os que apresentaram desempenhos diversificados para a classificação dos textos e eram compatíveis com a representação de documentos escolhida, uma vez que alguns métodos de classificação não trabalham com valores negativos nos vetores, característica presente no modelo de representação utilizado, Word2Vec. A diversidade no desempenho dos classificadores pôde ser identificada na criação da lista de *flags* (ver Seção 4.2), a qual é composta pelos resultados da avaliação dos classificadores para cada instância.

Para exemplificar a diversidade encontrada no desempenho dos classificadores, a Figura 5.1 apresenta como ficou a distribuição nas listas de *flags* para as bases de dados Slashdot e Ohsumed. Nessa figura, tem-se a quantidade de instâncias em que cada modelo de classificação foi escolhido como o mais apropriado. Além dos três métodos utilizados, tem-se também o comitê formado por eles (ver Tabela 4.1).

[6] https://code.google.com/archive/p/word2vec/.

Figura 5.1 – Distribuição nas listas de flags referentes as bases de dados Slashdot e Ohsumed, respectivamente.



Fonte: Elaborada pela autora (2020).

Os métodos utilizados como classificadores base para o método proposto foram: classificador Naive Bayes com o método de transformação Relevância Binária, Máquina de Vetores de Suporte (SVM) com o método de transformação *Label Powerset* (LP) e o classificador *Multilayer Perceptron* (MLP), o qual pode ser diretamente aplicado à problemas multirrótulo.

Utilizou-se a configuração padrão dos parâmetros dos classificadores, com exceção do classificador MLP, em que o algoritmo para otimização dos pesos foi configurado para o *'lbfgs'*, devido à agilidade na convergência.

5.1.5 Métrica de similaridade

Para encontrar a similaridade entre os textos, foram investigadas várias medidas de distância, similaridade e combinações de algumas. Foram realizados vários experimentos com diferentes medidas de similaridade e representações dos documentos. Porém, os melhores desempenhos foram obtidos com o emprego da métrica de similaridade do cosseno, a qual foi utilizada no método proposto.

Nesta seção foram descritas as principais ferramentas utilizadas na execução deste trabalho. Os conjuntos de dados utilizados nos experimentos foram detalhados e cada método referente à cada etapa do processo de classificação foi apresentado. Na sequência, serão apresentados os resultados experimentais e discussão.

5.2 Resultados experimentais e discussão

A avaliação do método proposto, denominado MetaMLC, tem como objetivo principal verificar se o método proposto é capaz de escolher o classificador mais apropriado para uma

instância de texto de um determinado problema de classificação multirrótulo. Para isso, o seu resultado de desempenho é comparado com os resultados de cada método de classificação base em cada conjunto de dados.

Japkowicz e Shah (2011) enfatizam e sugerem o uso de testes estatísticos para a análise da avaliação de algoritmos de aprendizagem para, dessa forma, ser possível evidenciar resultados ocorridos por acaso. Portanto, consoante aos autores, foram realizados testes estatísticos neste trabalho. Ainda segundo os autores, o teste mais apropriado para comparações entre dois classificadores em um único ou múltiplos domínios é o Wilcoxon. Esse teste, proposto por Wilcoxon (1992), é não paramétrico e utiliza métodos de ranqueamento para encontrar a significância estatística entre experimentos.

Os testes estatísticos foram realizados por meio da plataforma *web* STAC, desenvolvida por Rodríguez-Fdez et al. (2015). Essa plataforma é direcionada aos testes estatísticos para comparação de algoritmos. Nessa plataforma, o teste Wilcoxon assume que as medianas das diferenças entre dois grupos são iguais (hipótese nula). Dessa forma, os resultados dos testes podem rejeitar essa hipótese com um certo nível de significância α, o qual refere-se a probabilidade de rejeição da hipótese em um cenário verdadeiro. Como exemplo, para $\alpha = 0{,}05$ o nível de confiança do teste é de 95%. Nos testes realizados neste trabalho considerou-se $\alpha = 0{,}05$ em todas as comparações.

As Tabelas 5.2 e Tabela 5.3 apresentam os resultados da comparação do MetaMLC e de seus classificadores base usando as métricas de avaliação Micro F1 e *Hamming Loss*, respectivamente. Para cada conjunto de dados, foram realizadas três repetições experimentais com a validação cruzada de três partes. Dessa forma, o resultado de cada métrica é a média obtida a partir dessas três execuções e ao lado de cada resultado é apresentado o desvio padrão. Nessas tabelas, os valores em negrito indicam o melhor resultado obtido para cada conjunto de dados. Além disso, o símbolo (•) mostra que há uma diferença com significância estatística entre o classificador base em questão e o MetaMLC. Finalmente, a última linha apresenta as médias dos resultados obtidos por cada método considerando todos os conjuntos de dados.

De acordo com os resultados apresentados na Tabela 5.2, considerando a métrica Micro F1, o método proposto superou, com significância estatística, todos os classificadores base para os conjuntos de dados Slashdot, Ohsumed e Enron. Para os demais conjuntos de dados, o MetaMLC superou, com significância estatística, dois dos classificadores base e empatou com o terceiro.

Tabela 5.2 – Comparação dos resultados do desempenho dos classificadores com a métrica Micro F1.

Nome	MLP	SVM(LP)	NB(BR)	MetaMLC
G1	0,5296±0,0082 •	0,5438±0,0073 •	**0,5884±0,0062**	0,5804±0,0072
BFRC-PT	0,6648±0,0026	0,6324±0,0039 •	0,6125±0,0045 •	**0,6661±0,0028**
Medical	0,7269±0,0223 •	0,7599±0,0155	0,4546±0,0144 •	**0,7647±0,0202**
Slashdot	0,5283±0,0085 •	0,5848±0,0123 •	0,4158±0,0037 •	**0,5941±0,0104**
Ohsumed	0,5663±0,0054 •	0,5366±0,0053 •	0,3435±0,0028 •	**0,5737±0,0051**
Enron	0,5662±0,0054 •	0,4957±0,0060 •	0,2445±0,0065 •	**0,5726±0,0081**
Média	0,5970	0,5922	0,4432	**0,6252**

Fonte: Elaborada pela autora (2020).

Os resultados da avaliação dos classificadores para o conjunto de dados G1 apresentaram um comportamento bem distinto dos demais conjuntos de dados. O classificador NB(BR), que apresentou o menor desempenho dentre os classificadores base, para o conjunto de dados G1 ele obteve o melhor desempenho. No entanto, ao analisar o erro na predição no mesmo conjunto de dados usando a métrica *Hamming Loss* (veja Tabela 5.3), o MetaMLC o superou, estatisticamente, assim como a todos os outros classificadores base.

De uma forma geral, na avaliação com a métrica Micro F1, o objetivo do método proposto foi alcançado ao conseguir um desempenho médio geral superior aos obtidos pelos classificadores base. Por meio do teste estatístico realizado entre o MetaMLC e cada um de seus classificadores base, observou-se que o MetaMLC conseguiu ter um desempenho superior, com significância estatística, em cinco dos seis conjuntos de dados utilizados e empatou em um, comparando seu desempenho par a par com cada método base.

Conforme o apresentado na Tabela 5.3, considerando a avaliação do desempenho por meio da métrica *Hamming Loss*, o método proposto conseguiu superar, com significância estatística, os demais classificadores no conjunto de dados G1. Contudo, para os demais conjuntos de dados, o MetaMLC foi melhor que dois dos seus classificadores base e empatou com o terceiro. No entanto, na média geral obtida a partir do resultado do desempenho em todos os conjuntos de dados, o MetaMLC apresentou o melhor resultado.

Os testes estatísticos com a métrica *Hamming Loss*, comparando o MetaMLC e seus classificadores base par a par, mostraram que o MetaMLC superou estatisticamente o classificador NB(BR) em todas as bases de dados utilizadas. Em relação ao classificador SVM(LP), o MetaMLC teve um desempenho melhor estatisticamente em quatro das seis bases de dados utilizadas. Na comparação com o classificador MLP, o MetaMLC o superou, com significância estatística, em três bases de dados e empatou nas outras três.

Tabela 5.3 – Comparação dos resultados do desempenho dos classificadores com a métrica *Hamming Loss*.

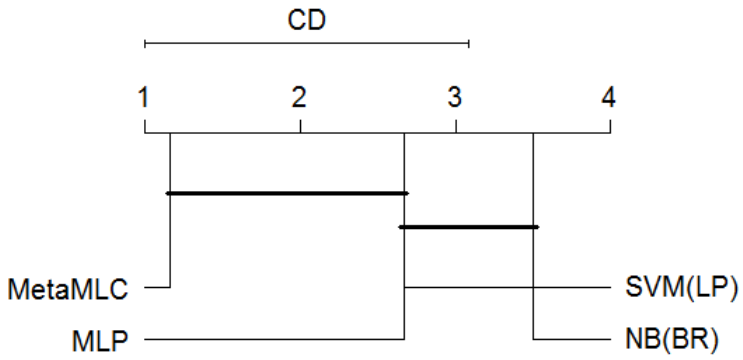
Nome	MLP	SVM(LP)	NB(BR)	MetaMLC
G1	0,2625±0,0039 •	0,2418±0,0033 •	0,2688±0,0045 •	**0,2310±0,0033**
BFRC-PT	**0,3111±0,0030**	0,3233±0,0034 •	0,4358±0,0051 •	0,3126±0,0023
Medical	0,0144±0,0011 •	**0,0125±0,0008**	0,0518±0,0028 •	**0,0125±0,0011**
Slashdot	0,0499±0,0011 •	**0,0415±0,0012**	0,1127±0,0019 •	0,0419±0,0010
Ohsumed	**0,0537±0,0008**	0,0563±0,0007 •	0,1900±0,0020 •	0,0540±0,0006
Enron	0,0506±0,0007	0,0567±0,0004 •	0,2941±0,0111 •	**0,0504±0,0011**
Média	0,1237	0,1220	0,2255	**0,1170**

Fonte: Elaborada pela autora (2020).

O MetaMLC também foi comparado aos classificadores base em todos os conjuntos de dados aplicando o teste de Friedman (FRIEDMAN, 1937), um teste não paramétrico para análise de desempenho algorítmico em vários conjuntos de dados. Por meio de ambas as métricas Micro F1 e *Hamming Loss*, de acordo com os resultados fornecidos pelo teste de Friedman, a hipótese nula (N_0), que pressupõe que as médias dos resultados de dois ou mais algoritmos são iguais, foi rejeitada com nível de significância de 0,05. Após constatada a diferença estatística entre os algoritmos, com o objetivo de comparar todos os métodos em todos os conjuntos de dados par a par, foi realizado o teste estatístico, sugerido por Demšar (2006), denominado Nemenyi (NEMENYI, 1963). Esse teste foi realizado no ambiente para computação estatística R (R Core Team, 2018). O teste Nemenyi é utilizado quando é detectada previamente diferença com significância estatística entre os métodos avaliados. Dado um nível de significância α, nesse caso 0,05, o teste Nemenyi determina um valor denominado diferença crítica, do inglês *Critical Difference* (CD), o qual é relacionado à diferença entre as médias do ranqueamento dos

classificadores. A diferença com significância estatística entre os classificadores é considerada quando essa diferença é maior ou igual ao valor de CD. O resultado do teste utilizando a métrica Micro F1 pode ser visualizado na Figura 5.2. Nessa figura, os métodos conectados pela barra em negrito são considerados estatisticamente equivalentes. O MetaMLC ocupou a melhor posição no ranqueamento e superou, com significância estatística, os métodos SVM(LP) e NB(BR), empatando com o método MLP.

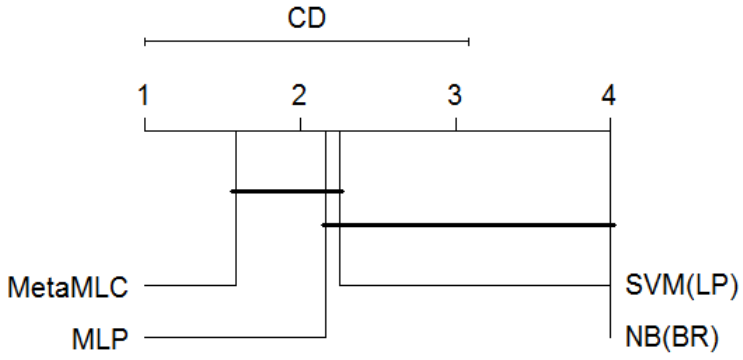
Figura 5.2 – *Critical Difference* (CD) para o teste Nemenyi com a métrica Micro F1.



Fonte: Elaborada pela autora (2020).

Um comportamento similar foi apresentado nos resultados do teste com a métrica *Hamming Loss* (ver Figura 5.3). O MetaMLC continuou em primeiro lugar no ranqueamento, porém, foi estatisticamente superior a um de seus classificadores base e empatou com os outros dois.

Figura 5.3 – *Critical Difference* (CD) para o teste Nemenyi com a métrica *Hamming Loss*.



Fonte: Elaborada pela autora (2020).

Os conjuntos de dados utilizados nos experimentos deste trabalho possuem domínios e características estatísticas bem distintos (veja Tabela 5.1), além de haver um desbalanceamento significativo na distribuição das classes de cada conjunto de dados. Portanto, a variação dos desempenhos dos classificadores é notável para cada conjunto de dados (ver Tabelas 5.2 e Tabela 5.3). Nesse contexto, é importante observar os benefícios do método proposto, que alcançou um desempenho médio superior ao obtido pelos classificadores base usando a seleção dinâmica de classificadores.

Este capítulo apresentou a análise experimental realizada neste trabalho. Para isso, inicialmente, foi abordada tanto a configuração experimental como os conjuntos de dados utilizados, métodos e ferramentas aplicadas desde o pré-processamento e representação dos conjuntos de dados até a avaliação dos resultados do desempenho dos classificadores. Em seguida, os resultados experimentais foram apresentados seguidos de uma discussão relacionada ao desempenho dos classificadores. O próximo capítulo apresenta as considerações finais deste trabalho com uma breve descrição da abordagem proposta, as contribuições do trabalho e possíveis investigações futuras.

6 CONSIDERAÇÕES FINAIS

Este capítulo apresenta a conclusão deste trabalho. Para tanto, a Seção 6.1 destaca a finalidade do método proposto e suas prerrogativas. Já a Seção 6.2 pontua as contribuições deste trabalho e a Seção 6.3 expõe o que pode ser desenvolvido como trabalhos futuros.

6.1 Visão geral do método proposto

A classificação multirrótulo ganhou atenção devido à sua importância em aplicações modernas e, quando se trata de textos, esse tipo de classificação permite encontrar várias classes às quais um texto pode estar associado. Vários algoritmos para a classificação multirrótulo estão disponíveis na literatura, no entanto, devido à variedade de desempenho em diferentes conjuntos de dados, é difícil escolher entre eles. Portanto, o presente trabalho investigou a aplicação da meta-aprendizagem na seleção dinâmica de classificadores. O objetivo principal foi o desenvolvimento de um método para a classificação de textos no contexto multirrótulo.

O método proposto seleciona entre um conjunto classificadores os que são mais adequados para cada instância do conjunto de treinamento e aplica esse conhecimento na fase de classificação, na qual a nova instância é classificada com base na sua vizinhança na base de dados de treinamento.

O método proposto foi comparado com os três métodos utilizados como base, usando duas bases de dados compostas por textos escritos no idioma português do Brasil e quatro bases de dados com textos escritos no idioma inglês. Para esta avaliação, foram adotadas duas métricas comumente usadas em problemas de classificação multirrótulo, Micro F1 e *Hamming Loss*. Os resultados experimentais mostraram que o método proposto superou os classificadores base na maioria dos conjuntos de dados e, portanto, é um método eficaz para a tarefa de classificação de texto multirrótulo.

6.2 Contribuições

A classificação de texto apresenta grande relevância na atualidade em consequência de grande parte do acentuado volume de dados gerados serem textos. Para que haja uma organização desses textos e que possa ser extraído conhecimento desses, são necessárias ferramentas automáticas.

A classificação multirrótulo faz-se necessária em problemas que a classificação tradicional não consegue solucionar. Tais problemas possuem como característica central a diversidade. Diante disso, este trabalho contribui com a indústria e o meio científico ao desenvolver um método para classificação de texto multirrótulo, no intuito de amenizar o trabalho do analista humano no momento de tomar a decisão sobre o classificador mais eficaz para aplicar ao seu problema de domínio textual. Em resumo, as contribuições deste trabalho incluem:

1. Desenvolvimento de um método para a classificação de texto no contexto multirrótulo por meio de seleção dinâmica de classificadores.

2. Exploração da meta-aprendizagem para seleção automática de classificadores.

6.3 Trabalhos futuros

Nos experimentos realizados, utilizamos apenas uma medida de similaridade para encontrar a semelhança entre os textos. Algumas outras foram investigadas, porém, experimentos adicionais são necessários para analisar se há eficácia no emprego das mesmas. O metaconhecimento explorado a partir do aprendizado base, consistiu na avaliação do desempenho dos classificadores. Dessa forma, há espaço para investigações futuras promissoras, tais como:

1. Investigar novos métodos de encontrar a similaridade entre textos de acordo com suas representações.

2. Explorar combinações de métricas de similaridade.

3. Explorar diferentes formas de meta-atributos, como características estatísticas dos conjuntos de dados.

4. Avaliar a aplicação da meta-aprendizagem ao explorar transferência de conhecimento entre domínios.

5. Avaliar cenários com bases de dados maiores e considerando fluxos de dados.

REFERÊNCIAS

AGGARWAL, C. C.; ZHAI, C. **Mining Text Data**. New York, USA: Springer-Verlag New York, 2012.

AMADO, A.; CORTEZ, P.; RITA, P.; MORO, S. Research trends on big data in marketing: A text mining and topic modeling based literature analysis. **European Research on Management and Business Economics**, v. 24, 07 2017.

ASSIS, M. S. de. Classificação multirrótulo com aprendizado semissupervisionado: uma análise multivisão de dados. **Dissertação de Mestrado**, Universidade Federal do Rio Grande do Norte, 2016.

BAEZA-YATES, R.; RIBEIRO-NETO, B. **Modern Information Retrieval: The Concepts and Technology behind Search**. England: Pearson Education Limited, 2011.

BALTE, A.; PISE, N.; KULKARNI, P. Meta-learning with landmarking: A survey. **International Journal of Computer Applications**, Citeseer, v. 105, n. 8, 2014.

BELL, J. **Machine learning: hands-on for developers and technical professionals**. Indianapolis, IN, USA: John Wiley & Sons, 2014.

BRAMER, M. **Principles of Data Mining**. London, England: Springer Publishing Company, Incorporated, 2013.

BRAZDIL, P.; CARRIER, C. G.; SOARES, C.; VILALTA, R. **Metalearning: Applications to data mining**. Berlin Heidelberg: Springer-Verlag, 2009.

BRAZDIL, P.; GIRAUD-CARRIER, C. Metalearning and algorithm selection: Progress, state of the art and introduction to the 2018 special issue. **Machine Learning**, v. 107, n. 1, jan. 2018.

BREIMAN, L.; FRIEDMAN, J.; OLSHEN, R.; STONE, C. Classification and regression trees. wadsworth int. v. 37, n. 15, p. 237–251, 1984.

CHEKINA, L.; ROKACH, L.; SHAPIRA, B. Meta-learning for selecting a multi-label classification algorithm. In: **Proceedings of the 2011 IEEE 11th International Conference on Data Mining Workshops (ICDMW)**. Washington, DC, USA: IEEE Computer Society, 2011. p. 220–227.

CHOPRA, A.; PRASHAR, A.; SAIN, C. Natural language processing. **International Journal of Technology Enhancements and Emerging Engineering Research**, v. 1, n. 4, p. 131–134, 2013.

CICHOSZ, P. **Data Mining Algorithms: Explained Using R**. Chichester, UK: John Wiley & Sons, 2014.

CLARE, A.; KING, R. D. Knowledge discovery in multi-label phenotype data. In: SPRINGER. **Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery**. Berlin, Heidelberg, 2001. p. 42–53.

CRUZ, R. M. O.; SABOURIN, R.; CAVALCANTI, G. D.; REN, T. I. Meta-des: A dynamic ensemble selection framework using meta-learning. **Pattern recognition**, New York, USA, v. 48, n. 5, p. 1925–1935, 2015.

CURI, Z.; BRITTO JR., A. d. S.; PARAISO, E. C. Multi-label classification of user reactions in online news. **arXiv preprint arXiv:1809.02811**, 2018.

DEAN, J. **Big data, data mining, and machine learning: value creation for business leaders and practitioners**. Hoboken, NJ, USA: John Wiley & Sons, 2014.

DEMsAR, J. Statistical comparisons of classifiers over multiple data sets. **Journal of Machine Learning Research**, JMLR.org, v. 7, p. 1–30, dez. 2006.

DO, C. B.; NG, A. Y. Transfer learning for text classification. In: **Proceedings of the 18th International Conference on Neural Information Processing Systems**. Cambridge, MA, USA: MIT Press, 2005. p. 299–306.

FERREIRA, T. G. Nicesim: um simulador interativo baseado em técnicas de aprendizado de máquina para avaliação de recém-nascidos prematuros em uti neonatal. **Dissertação de Mestrado**, Universidade Federal de Viçosa, 2014.

FRIEDMAN, M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. **Journal of the american statistical association**, Taylor & Francis, v. 32, n. 200, p. 675–701, 1937.

GALLINUCCI, E.; GOLFARELLI, M.; RIZZI, S. Advanced topic modeling for social business intelligence. **Information Systems**, Elsevier, v. 53, p. 87–106, 2015.

HAN, J.; KAMBER, M.; PEI, J. **Data Mining: Concepts and Techniques**. Waltham, MA, USA: Morgan Kaufmann Publishers Inc., 2011.

HARTMANN, N.; FONSECA, E.; SHULBY, C.; TREVISO, M.; SILVA, J.; ALUÍSIO, S. Portuguese word embeddings: Evaluating on word analogies and natural language tasks. In: **Proceedings of the 11th Brazilian Symposium in Information and Human Language Technology**. Uberlândia, Brazil: Sociedade Brasileira de Computação, 2017. p. 122–131.

HOTHO, A.; NÜRNBERGER, A.; PAASS, G. A brief survey of text mining. **LDV Forum**, v. 20, p. 19–62, 2005.

IRFAN, R.; KING, C. K.; GRAGES, D.; EWEN, S.; KHAN, S. U.; MADANI, S. A.; KOLODZIEJ, J.; WANG, L.; CHEN, D.; RAYES, A. et al. A survey on text mining in social networks. **The Knowledge Engineering Review**, v. 30, n. 2, p. 157–170, 2015.

JAPKOWICZ, N.; SHAH, M. **Evaluating learning algorithms: a classification perspective**. New York, USA: Cambridge University Press, 2011.

KASS, G. V. An exploratory technique for investigating large quantities of categorical data. **Journal of the Royal Statistical Society. Series C (Applied Statistics)**, [Wiley, Royal Statistical Society], v. 29, n. 2, p. 119–127, 1980.

LEMKE, C.; BUDKA, M.; GABRYS, B. Metalearning: a survey of trends and technologies. **Artificial intelligence review**, v. 44, n. 1, p. 117–130, 2015.

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

MIKOLOV, T.; CHEN, K.; CORRADO, G.; DEAN, J. Efficient estimation of word representations in vector space. **arXiv preprint arXiv:1301.3781**, 2013.

MIKOLOV, T.; SUTSKEVER, I.; CHEN, K.; CORRADO, G.; DEAN, J. Distributed representations of words and phrases and their compositionality. In: **Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2**. USA: Curran Associates Inc., 2013. (NIPS'13), p. 3111–3119.

MORO, S.; RITA, P.; VALA, B. Predicting social media performance metrics and evaluation of the impact on brand building: A data mining approach. **Journal of Business Research**, v. 69, p. 11, 2016.

NEMENYI, P. Distribution-free multiple comparisons. Tese de Doutorado, Princeton University, 1963.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, E. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011.

PEREIRA, R. B.; PLASTINO, A.; ZADROZNY, B.; MERSCHMANN, L. H. Correlation analysis of performance measures for multi-label classification. **Information Processing & Management**, v. 54, n. 3, p. 359–369, 2018.

PESTIAN, J. P.; BREW, C.; MATYKIEWICZ, P.; HOVERMALE, D. J.; JOHNSON, N.; COHEN, K. B.; DUCH, W. A shared task involving multi-label classification of clinical free text. In: **Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing**. Stroudsburg, PA, USA: Association for Computational Linguistics, 2007. p. 97–104.

QUINLAN, J. R. **C4.5: Programs for Machine Learning**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

R Core Team. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria, 2018. Disponível em: <https://www.R-project.org/>.

READ, J.; PFAHRINGER, B.; HOLMES, G. Multi-label classification using ensembles of pruned sets. In: **Proceedings of the 2008 Eighth IEEE International Conference on Data Mining (ICDM)**. Washington, DC, USA: IEEE Computer Society, 2008. p. 995–1000.

READ, J.; PFAHRINGER, B.; HOLMES, G.; FRANK, E. Classifier chains for multi-label classification. **Machine Learning Journal**, v. 85, n. 3, p. 333–359, 2011.

READ, J.; REUTEMANN, P.; PFAHRINGER, B.; HOLMES, G. MEKA: A multi-label/multi-target extension to Weka. **Journal of Machine Learning Research**, v. 17, n. 21, p. 1–5, 2016.

RODRíGUEZ-FDEZ, I.; CANOSA, A.; MUCIENTES, M.; BUGARíN, A. STAC: a web platform for the comparison of algorithms using statistical tests. In: IEEE. **Proceedings of the 2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)**. Istanbul, Turkey, 2015. p. 1–8.

ROKACH, L.; MAIMON, O. **Data mining with decision trees: theory and applications**. Singapore: World scientific Publishing Co. Pte. Ltd., 2014.

RUSSELL, S. J.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. 3th. ed. Upper Saddle River, NJ, USA: Prentice Hall, 2010.

SÁ, A. G. C. de; PAPPA, G. L.; FREITAS, A. A. Towards a method for automatically selecting and configuring multi-label classification algorithms. In: **Proceedings of the Genetic and Evolutionary Computation Conference Companion(GECCO 2017)**. Berlin, Germany: ACM, 2017. p. 1125–1132.

SÁ, A. G. de; FREITAS, A. A.; PAPPA, G. L. Automated selection and configuration of multi-label classification algorithms with grammar-based genetic programming. In: **15th International Conference on Parallel Problem Solving from Nature**. Coimbra, Portugal: Springer, 2018. p. 308–320.

SECHIDIS, K.; TSOUMAKAS, G.; VLAHAVAS, I. On the stratification of multi-label data. In: **Proceedings of the 2011 European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part III**. Berlin, Heidelberg: Springer-Verlag, 2011. (ECML PKDD'11), p. 145–158.

SILVA, J.; BRANCO, A.; CASTRO, S.; REIS, R. Out-of-the-box robust parsing of portuguese. In: **Proceedings of the 9th International Conference on the Computational Processing of Portuguese (PROPOR'10)**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 75–85.

SILVA, P. N. D. Classificação multirrótulo em cadeia: Novas abordagens. Dissertação de Mestrado, Universidade Federal Fluminense, 2014.

SZYMAŃSKI, P.; KAJDANOWICZ, T. A network perspective on stratification of multi-label data. In: **First International Workshop on Learning with Imbalanced Domains: Theory and Applications**. ECML-PKDD, Skopje, Macedonia: PMLR, 2017. p. 22–35.

SZYMAŃSKI, P.; KAJDANOWICZ, T. Scikit-multilearn: a scikit-based python environment for performing multi-label classification. **The Journal of Machine Learning Research**, JMLR. org, v. 20, n. 1, p. 209–230, 2019.

TENENBOIM-CHEKINA, L.; ROKACH, L.; SHAPIRA, B. Identification of label dependencies for multi-label classification. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING(ICML). **Working Notes of the Second International Workshop on Learning from Multi-Label Data**. Haifa, Israel, 2010. p. 53–60.

TRIPATHI, N.; OAKES, M.; WERMTER, S. A scalable meta-classifier combining search and classification techniques for multi-level text categorization. **International Journal of Computational Intelligence and Applications**, v. 14, n. 04, p. 1550020:1–1550020:16, 2015.

TSOUMAKAS, G.; KATAKIS, I. Multi-label classification: An overview. **International Journal of Data Warehousing and Mining (IJDWM)**, IGI Global, v. 3, n. 3, p. 1–13, 2007.

TSOUMAKAS, G.; KATAKIS, I.; VLAHAVAS, I. Effective and efficient multilabel classification in domains with large number of labels. In: ECML/PKDD 2008. **Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, Workshop on Mining Multidimensional Data (MMD'08)**. Antwerp, Belgium, 2008. v. 21, p. 53–59.

TSOUMAKAS, G.; VLAHAVAS, I. P. Random k -labelsets: An ensemble method for multilabel classification. In: **Machine Learning: ECML 2007, 18th European Conference on Machine Learning**. Warsaw, Poland: Springer, 2007. p. 406–417.

UYSAL, A. K.; GUNAL, S. The impact of preprocessing on text classification. **Information Processing & Management**, v. 50, n. 1, p. 104–112, 2014.

VAPNIK, V. N. An overview of statistical learning theory. **IEEE transactions on neural networks**, v. 10, n. 5, p. 988–999, 1999.

WEISS, S. M.; INDURKHYA, N.; ZHANG, T. **Fundamentals of predictive text mining**. London, UK: Springer-Verlag London, 2015.

WILCOXON, F. Individual comparisons by ranking methods. In: KOTZ, S.; JOHNSON, N. L. (Ed.). **Breakthroughs in Statistics: Methodology and Distribution**. New York, NY: Springer New York, 1992. p. 196–202.

WITTEN, I. H.; FRANK, E.; HALL, M. A.; PAL, C. J. **Data Mining: Practical machine learning tools and techniques**. Burlington, MA, USA: Morgan Kaufmann, 2016.

ZHANG, L.; JIANG, L.; LI, C. A discriminative model selection approach and its application to text classification. **Neural Computing and Applications**, p. 1–15, 2017.

ZHANG, M.; ZHOU, Z. A k-nearest neighbor based algorithm for multi-label classification. In: IEEE. **2005 IEEE International Conference on Granular Computing,**. Beijing, China, 2005. p. 718–721.

ZHANG, M.-L.; LI, Y.-K.; LIU, X.-Y.; GENG, X. Binary relevance for multi-label learning: an overview. **Frontiers of Computer Science**, Springer, p. 1–12, 2018.

ZHANG, M.-L.; ZHOU, Z.-H. A review on multi-label learning algorithms. **IEEE transactions on knowledge and data engineering**, v. 26, n. 8, p. 1819–1837, 2014.