UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

AIRTON BORDIN JUNIOR

Aplicação de Programação Genética na Análise de Sentimentos

Goiânia
2018

sistema de bibliotecas ufg

PRPG
PRÓ-REITORIA DE PÓS-GRADUAÇÃO

UFG

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TESES E DISSERTAÇÕES NA BIBLIOTECA DIGITAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio da Biblioteca Digital de Teses e Dissertações (BDTD/UFG), regulamentada pela Resolução CEPEC nº 832/2007, sem ressarcimento dos direitos autorais, de acordo com a Lei nº 9610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou *download*, a título de divulgação da produção científica brasileira, a partir desta data.

1. Identificação do material bibliográfico: **[X] Dissertação** **[ ] Tese**

2. Identificação da Tese ou Dissertação:

Nome completo do autor: Airton Bordin Junior

Título do trabalho: Aplicação de Programação Genética na Análise de Sentimentos

3. Informações de acesso ao documento:

Concorda com a liberação total do documento [X] SIM [ ] NÃO[1]

Havendo concordância com a disponibilização eletrônica, torna-se imprescindível o envio do(s) arquivo(s) em formato digital PDF da tese ou dissertação.

Assinatura do(a) autor(a)[2]
Airton Bordin Junior

Ciente e de acordo:

Assinatura do(a) orientador(a)[2]
Nádia Félix Felipe da Silva

Data: 17 / 12 / 2018

AIRTON BORDIN JUNIOR

Aplicação de Programação Genética na Análise de Sentimentos

Dissertação apresentada ao Programa de Pós–Graduação do Instituto de Informática da Universidade Federal de Goiás, como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Ciência da Computação

Orientadora: Profa. Nádia Félix Felipe da Silva

Coorientador: Prof. Celso Gonçalves Camilo Junior

Goiânia
2018

Ficha de identificação da obra elaborada pelo autor, através do
Programa de Geração Automática do Sistema de Bibliotecas da UFG.

Bordin Junior, Airton
Aplicação de Programação Genética na Análise de Sentimentos [manuscrito] / Airton Bordin Junior. - 2018.
CXLII, 142 f.

Orientador: Profa. Dra. Nádia Félix Felipe da Silva; co-orientador Dr. Celso Gonçalves Camilo Junior.
Dissertação (Mestrado) - Universidade Federal de Goiás, Instituto de Informática (INF), Programa de Pós-Graduação em Ciência da Computação, Goiânia, 2018.

Inclui siglas, abreviaturas, símbolos, gráfico, tabelas, algoritmos, lista de figuras, lista de tabelas.

1. Análise de Sentimentos. 2. Mineração de Opiniões. 3. Programação Genética. 4. Classificadores. I. Félix Felipe da Silva, Nádia , orient. II. Título.

CDU 004

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

UFG

ATA Nº 25/2018

ATA DA SESSÃO DE JULGAMENTO DA DISSERTAÇÃO DE MESTRADO DE AIRTON BORDIN JÚNIOR

Aos catorze dias do mês de dezembro de dois mil e dezoito, às dezenove horas, na sala 150 do Instituto de Informática da Universidade Federal de Goiás, Campus Samambaia, reuniu-se a banca examinadora designada na forma regimental pela Coordenação do Curso para julgar a dissertação de mestrado intitulada **"Aplicação de Programação Genética na Análise de Sentimentos"**, apresentada pelo aluno Airton Bordin Júnior como parte dos requisitos necessários à obtenção do grau de Mestre em Ciência da Computação, área de concentração Ciência da Computação. A banca examinadora foi presidida pela orientadora do trabalho de dissertação, Professora Doutora Nádia Félix Felipe da Silva (INF/UFG), tendo como membros os Professores Doutores Celso Gonçalves Camilo Júnior (INF/UFG – coorientador), Thierson Couto Rosa (INF/UFG) e Thiago Ferreira Covões (UFABC). O prof. Thiago Covões participou a distância por webconferência. Aberta a sessão, o candidato expôs seu trabalho. Em seguida, o aluno foi arguido pelos membros da banca e:

(X) tendo demonstrado suficiência de conhecimento e capacidade de sistematização do tema de sua dissertação, a banca concluiu pela **aprovação** do candidato, sem restrições.

( ) não tendo demonstrado suficiência de conhecimento e capacidade de sistematização do tema de sua dissertação, a banca concluiu pela **reprovação** do candidato.

Os trabalhos foram encerrados às 21 horas. Nos termos do Regulamento Geral dos Cursos de Pós-Graduação desta Universidade, lavrou-se a presente ata que, lida e julgada conforme, segue assinada pelos membros da banca examinadora.

Profa. Dra. Nádia Félix Felipe da Silva Nádia Félix F. da Silva

Prof. Dr. Celso Gonçalves Camilo Júnior Celso Camilo

Prof. Dr. Thierson Couto Rosa Thierson Couto Rosa

Prof. Dr. Thiago Ferreira Covões Thiago Ferreira Covões

Airton Bordin Junior

Bacharel em Ciência da Computação pela Faculdade Anglo-Americano, mestrando em Informática pela Universidade Federal de Goiás. Possui especialização *Lato Sensu* em Redes de Computadores. Bolsista do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

Dedico este trabalho a Deus, pelas bênçãos recebidas, à minha mãe, que lá do céu ilumina o meu caminho e me ajuda a seguir em frente, à minha avó, que sempre me orienta e me dá forças para conquistar os meus sonhos, ao meu pai, que é muito especial para mim, a todas as pessoas da minha família, pelo imenso carinho que nos une apesar da distância física que nos separa, e ao meu amor, que está ao meu lado em todos os momentos.

Agradecimentos

A Deus, à minha família e ao meu amor, que são os meus alicerces, as razões das minhas conquistas e a paz que preciso para imergir no universo de conhecimento que o mestrado me proporcionou. Aos amigos que fiz em Foz do Iguaçu, em Goiânia e aos que conheci em outros locais e circunstâncias, mas que seguem como parte da minha vida.

À minha orientadora, Nádia Félix Felipe da Silva, que acreditou em mim, confiou no nosso projeto e, com extrema dedicação e conhecimento, conduziu-me à conclusão deste mestrado. Ao meu coorientador, Celso Gonçalves Camilo Júnior, que com sabedoria me ajudou a encontrar o melhor caminho para esta pesquisa. Ao professor Thierson Couto Rosa que, em conjunto com a minha orientadora, acompanhou-me semanalmente com sugestões imprescindíveis para a estruturação da minha pesquisa.

À professora Telma Woerle de Lima Soares, que esteve presente na minha qualificação e me sugeriu mudanças fundamentais para o resultado alcançado. Aos professores Deborah Silva Alves Fernandes e Thiago Ferreira Covões que, prontamente, aceitaram o convite para compor a minha banca de defesa.

Aos professores e amigos do mestrado, que com orientações valiosas me ajudaram a aperfeiçoar minha pesquisa e a concretizar o sonho de concluir este projeto de vida. A todos os colegas do Instituto de Informática da Universidade Federal de Goiás (INF/UFG), pelo apoio e presteza em todos os momentos.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), pelo suporte financeiro durante o mestrado.

The original question, “Can machines think?” I believe to be too meaningless to deserve discussion.

Alan Turing,
Computing Machinery and Intelligence.

Resumo

<Bordin Junior, Airton>. **Aplicação de Programação Genética na Análise de Sentimentos**. Goiânia, 2018. 143p. Dissertação de Mestrado. Instituto de Informática, Universidade Federal de Goiás.

A *Web* é comumente utilizada como plataforma para debates, opiniões, avaliações, etc. Esses dados permitiram que a área de Análise de Sentimentos (AS) se desenvolvesse para extrair informações e conhecimentos que possam ser utilizados em diferentes aplicações. Entre os desafios da AS, destacam-se a criação de classificadores com boa eficácia. Normalmente, os modelos de classificação gerados são heurísticas específicas, manualmente definidas e pouco adaptáveis a diferentes contextos. Assim, o presente trabalho propõe a geração automatizada de classificadores de sentimentos híbridos – utilizando técnicas de Aprendizado de Máquina (AM) e dicionários léxicos – com o uso da Programação Genética (PG). Com isso, espera-se reduzir o custo de geração dos classificadores e aumentar o poder de predição para cada domínio analisado. A intenção é que esses classificadores sejam competitivos com os algoritmos clássicos empregados na área de AS, generalizáveis, adaptáveis ao contexto e capazes de determinar a relevância de cada um dos dicionários léxicos ao domínio aplicado. Além disso, a ideia é que seja possível a agregação de outras técnicas de AM para a geração de soluções híbridas ainda mais eficazes. Para validar a proposta, foi utilizado o *benchmark* SemEval 2014 e os resultados mostram que a abordagem de geração automatizada com a PG é promissora, pois os modelos gerados são competitivos e, algumas vezes, superiores aos de outros trabalhos da literatura. A combinação dos classificadores em um comitê mostrou-se eficaz ao aumento do poder de predição do sistema, obtendo resultados superiores à utilização das técnicas individualmente. Por fim, destaca-se a capacidade de customização dos modelos de acordo com o contexto abordado e a possibilidade de transferência de conhecimento dos usuários por meio das funções utilizadas pela PG.

Palavras–chave

Análise de Sentimentos, Mineração de Opiniões, Programação Genética, Classificadores

Abstract

<Bordin Junior, Airton>. **Applying Genetic Programming to Sentiment Analysis**. Goiânia, 2018. 143p. MSc. Dissertation. Instituto de Informática, Universidade Federal de Goiás.

The Web is commonly used as a platform for debates, opinions, evaluations, etc. These data allowed the area of Sentiment Analysis (SA) to develop to extract information and knowledge that can be used in different applications. Among the challenges of SA we can highlight the creation of classifiers with good efficacy. Typically, the classification models are generated using specific heuristics, manually defined and not adaptable to different contexts. Thus, this work proposes the automated generation of hybrid SA classifiers - with Machine Learning (ML) techniques and lexical dictionaries - using Genetic Programming (GP). It is expected to reduce the cost of generating the classifiers and increase the predictive power for each domain analyzed. The goal is that these classifiers will be competitive with the classical ML algorithms used in SA, generalizable, adaptable to the context and able to determine the relevance of each lexical to the applied domain. In addition, the aim is allow to aggregate other ML techniques to create even more effective hybrid solutions. In order to validate the proposal, SemEval 2014 benchmark was used. The results show that the approach with GP is promising since the generated models are competitive, and sometimes better, with other researches. The ensemble proved to be effective in increasing the predictive power of the system, obtaining better results than the use of the techniques individually. Finally, we highlight the ability of models customization according to the context approached and the possibility of knowledge transfer of the users through the functions used by GP.

Keywords

Sentiment Analysis, Opinion Mining, Genetic Programming, Classifiers

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

Lista de Siglas

AG Algoritmos Genéticos.
AM Aprendizado de Máquina.
AO Análise de Opiniões.
AS Análise de Sentimentos.
AST Análise de Sentimentos em *Tweets*.

BOW *Bag-of-words*.

CRF *Conditional Random Field*.

DEAP *Distributed Evolutionary Algorithms in Python*.

MO Mineração de Opiniões.

NB *Naïve Bayes*.

PG Programação Genética.
PGFT Programação Genética Fortemente Tipada.
PGS Programação Genética Semântica.
PLN Processamento de Linguagem Natural.
PMI *Pointwise Mutual Infomation*.
POS *Part-of-Speech*.
PSO *Particle Swarm Optimization*.

RF *Random Forest*.
RGP *Root Genetic Programming*.
RL Regressão Logística.
RNA Redes Neurais Artificiais.

SGD *Stochastic Gradient Descent*.
SVM *Support Vector Machines*.

TF-IDF *Term Frequency – Inverse Document Frequency*.

CAPÍTULO 1

Introdução

A Análise de Sentimentos (AS) é uma linha de pesquisa que tem por objetivo classificar as emoções de um determinado texto, geralmente em positivo, negativo ou neutro [104]. A área vem ganhando destaque nos últimos anos, principalmente por conta da popularização do acesso à Internet e do consequente aumento na quantidade de conteúdo produzido na rede. O uso das redes sociais, como *Twitter*[1], Facebook[2] e Instagram[3], e a forma como os usuários compartilham suas opiniões e sentimentos sobre os mais diversos assuntos, têm motivado a pesquisa de classificadores para esses conteúdos.

As abordagens de classificação de sentimentos são comumente divididas em três classes principais [52, 68, 185]: técnicas que utilizam Aprendizado de Máquina (AM), baseadas em Léxico e Híbridas. A primeira delas usa abordagens de Aprendizado de Máquina supervisionadas para a classificação das opiniões, realizando o treinamento com mensagens previamente rotuladas. As abordagens Léxicas são heurísticas baseadas em aspectos estruturais do texto e que empregam dicionários léxicos – conjunto de palavras e suas polaridades, frequentemente definidas como positiva ou negativa – para a avaliação do texto. As técnicas híbridas utilizam as duas abordagens anteriores de forma conjunta.

Para que um classificador tenha resultados generalizáveis e bom poder de predição, deve levar em consideração aspectos inerentes ao contexto das opiniões que serão avaliadas. Um modelo de classificação de *tweets* (mensagens postadas no *Twitter*), por exemplo, geralmente é diferente de um processo de classificação de avaliações de produtos ou comentários políticos. No primeiro, por conta da limitação do tamanho das mensagens em 280 caracteres, o uso de abreviações é muito comum. Além disso, por se tratar de uma plataforma de rede social, os textos são frequentemente escritos de maneira informal, comumente fazendo uso de gírias [59].

Classificadores de sentimentos são gerados, frequentemente, de forma manual.

[1] https://twitter.com/
[2] https://www.facebook.com/
[3] https://instagram.com

Essa criação é feita por especialistas no domínio específico da classificação, capturando, dessa forma, a experiência do projetista sobre o contexto analisado [21, 85, 68, 84]. No entanto, isso aumenta o custo de geração para cada cenário, além de penalizar a generalização [16].

Com o intuito de automatizar essa tarefa, o desafio de geração de um classificador de sentimento pode ser formulado como um problema de busca e otimização, que tem como objetivo encontrar um modelo – dentro do espaço de modelos possíveis – que maximize a quantidade de avaliações corretas. Abordando dessa forma, pode-se utilizar qualquer método de Aprendizado de Máquina que retorne um classificador como resultado [106].

Entre os métodos de AM existentes utilizados para geração de modelos, a Programação Genética (PG) é uma das alternativas. A PG é uma abordagem da área da computação evolucionária que busca a criação automatizada de modelos baseados na função objetivo, que representa o problema [157, 108]. Uma das vantagens das soluções descobertas pela PG em relação a outras técnicas de AM é que elas podem ser lidas e interpretadas pelos usuários, ou seja, pode-se analisar os resultados de forma a entender o processo da solução [96]. Essas soluções são frequentemente representadas por meio de árvores.

Diferentemente de outros métodos tradicionais, como as Redes Neurais Artificiais (RNA), a PG gera uma representação transparente e estruturada do sistema, e não exige um conhecimento antecipado da estrutura geral do modelo a ser gerado. Essa solução difere consideravalmente da RNA, por exemplo, onde a estrutura da rede deve ser estabelecida no início do processo, e os ciclos de evolução dos modelos aprimoram somente os atributos dessa estrutura, como pesos.

Na PG, tanto os parâmetros quanto a estrutura geral são parte do processo de otimização, pois integram processo de busca [136]. Essas características permitem que a PG identifique os valores que contribuem de forma positiva para o modelo, desconsiderando os que não possuírem valor significativo durante a evolução, reduzindo de forma considerável a complexidade dos modelos [136].

Além disso, com relação às vantagens da utilização da Programação Genética, é possível citar a simplicidade conceitual da abordagem, a sua ampla aplicabilidade (podendo ser aplicada a virtualmente qualquer problema de busca e otimização), o seu alto grau de paralelismo, flexibilidade, possibilidade de utilização com pouco conhecimento sobre o problema em questão, entre outras [56].

No contexto da AS, podem-se destacar alguns benefícios da utilização da PG. A forma de representação dos modelos, conforme citado anteriormente, facilita a leitura do processo de classificação das mensagens. Essa característica permite expandir o entendimento sobre a AS no contexto da análise, bem como a validação de alguma

hipótese sobre o classificador final. Em relação a outras técnicas de AM, como o *Deep Learning*, por exemplo, a PG demanda uma quantidade consideravelmente menor de dados para a geração de modelos, facilitando a utilização em diversos domínios, mesmo com um *dataset* consideravelmente pequeno disponível [13].

Considerando a importância da utilização e combinação de léxicos nas soluções de AS [11], a abordagem por meio da PG facilita o arranjo desses dicionários na solução, auxiliando na definição da importância e peso de cada um deles para a avaliação geral das mensagens em determinado domínio. Isso é importante, pois a simples inserção de mais dicionários não reflete necessariamente em melhores resultados e é preciso a adequação desses recursos ao contexto da análise em questão.

Com vista ao exposto nos parágrafos anteriores, este trabalho tem como objetivo principal gerar, de forma automatizada, classificadores de sentimentos híbridos – utilizando técnicas de Aprendizado de Máquina e dicionários léxicos – com o uso da Programação Genética. A intenção é que esses classificadores sejam competitivos com os algoritmos clássicos empregados na área de AS, adaptáveis ao contexto e capazes de determinar a relevância de cada um dos dicionários léxicos ao domínio aplicado. Além disso, o intuito é que seja possível a agregação de outras técnicas de AM para a geração de classificadores híbridos ainda mais eficazes.

Para atingir o objetivo supracitado, duas frentes principais de trabalho foram definidas: a primeira delas busca a criação de um processo de geração automatizada de classificadores híbridos de sentimentos utilizando a PG como técnica principal, além do uso de dicionários léxicos disponíveis na literatura. Com isso, as seguintes hipóteses de pesquisa foram levantadas:

Hipótese de pesquisa 1: Classificadores híbridos de sentimentos inferidos automaticamente com o uso de Programação Genética apresentam resultados competitivos ou superiores aos valores obtidos a partir de técnicas clássicas de Aprendizado de Máquina.

Hipótese de pesquisa 2: Classificadores híbridos de sentimentos inferidos por meio de Programação Genética podem ser usados para, a partir de vários léxicos disponíveis na literatura, aprender e escolher quais são os mais relevantes para o domínio em questão.

Para a validação das duas primeiras hipóteses definidas, faz-se necessário a utilização de outras técnicas clássicas de Aprendizado de Máquina, de forma a validar a competitividade da abordagem, como levantado na Hipótese de Pesquisa 1. A segunda frente de trabalho, portanto, resulta da utilização desses algoritmos clássicos, juntamente com a PG, em um esquema de comitê de classificadores [194]. Esses agrupamentos podem incrementar o poder de predição do sistema, alcançando valores superiores aos

atingidos individualmente pelas técnicas, caso possuam a propriedade de diversidade [41, 160, 37]. Com isso, a terceira Hipótese de Pesquisa é definida:

Hipótese de pesquisa 3: A utilização da PG, em conjunto com outras técnicas de Aprendizado de Máquina, organizados em um comitê de classificadores, pode incrementar o poder de predição e o resultado geral do sistema de AS, resultando em valores superiores aos alcançados individualmente pelas técnicas.

Os testes mostraram que as abordagens utilizando Programação Genética podem ser consideradas competitivas quando comparadas às diferentes técnicas tradicionais de Aprendizado de Máquina – *Support Vector Machines* (SVM), *Naïve Bayes*, *Random Forest*, Regressão Logística e *Stochastic Gradient Descent* (SGD) – e com outros trabalhos da literatura, conforme exposto em detalhes no Capítulo 6. Vale destacar que a versão atualizada da PG – chamada de PG_a – obteve os melhores resultados, mostrando que uma alteração no processo de treinamento da PG foi capaz de aperfeiçoar a generalização e, consequentemente, melhorar o poder preditivo do modelo. Por fim, a combinação dos classificadores foi eficaz para o aumento do poder de predição da solução, especialmente a combinação que inclui a PG – chamada de $ensemble_{pg}$ – que obteve resultados superiores aos alcançados individualmente pelas técnicas.

Para facilitar o entendimento da solução proposta, este trabalho está organizado da seguinte forma: inicialmente, conceitos essenciais para o entendimento do problema de pesquisa são apresentados no Capítulo 2. Na sequência, os trabalhos relacionados são discutidos no Capítulo 3. A proposta e o fluxo geral deste trabalho são apresentados no Capítulo 4. Os experimentos são descritos no Capítulo 5 e os resultados da pesquisa são apresentados no Capítulo 6. Por fim, o Capítulo 7 discorre sobre as considerações finais acerca da pesquisa e discute alguns possíveis trabalhos futuros para a melhoria dos resultados.

CAPÍTULO 2

Fundamentação Teórica

Neste Capítulo, são exibidos os principais conceitos necessários para a compreensão da pesquisa realizada. A Seção 2.1 apresenta uma descrição da Análise de Sentimentos e seus principais desafios e aplicações, bem como as definições fundamentais utilizadas ao longo do trabalho. A Seção 2.2 discorre sobre a Programação Genética, suas principais características e áreas nas quais vem sendo empregada com sucesso. A Seção 2.3 apresenta uma visão geral sobre a técnica de Comitês de Classificadores e suas principais abordagens.

2.1 Análise de Sentimentos

A Análise de Sentimentos (AS) – também chamada de Análise de Opiniões (AO) ou Mineração de Opiniões (MO) – é uma linha de pesquisa abrangente, considerada uma subárea da área de Processamento de Linguagem Natural (PLN), e que vem sendo tema de diversos trabalhos nos últimos anos. Apesar da PLN ser alvo de inúmeras pesquisas e aplicações há algumas décadas, pouca atenção foi dada à AS até o ano 2000 [141].

O crescente interesse sobre o assunto decorre, principalmente, do aumento no número de usuários da Internet e sua consolidação como importante plataforma para difusão de conteúdo independente como debates, opiniões, avaliações, entre outros [104].

Cada vez mais pessoas utilizam a Internet para criar conteúdo, compartilhar rotinas, avaliar produtos, dar opiniões sobre os mais diversos assuntos, etc. Redes sociais, como o *Facebook* e o *Twitter*, tornaram-se importantes agregadores de tópicos, gerando uma grande quantidade de dados. Esses, por sua via, são extremamente valiosos e podem servir como direcionamento e apoio na tomada de decisões de empresas, governos, outros usuários, etc.

Nesse contexto, um dos principais desafios da AS é a Análise de Opiniões, descritas em linguagem natural, para a identificação da polaridade implícita ou explícita no texto. Essa polaridade é, na maior parte das vezes, identificada como uma escala de pontuação de sua característica positiva ou negativa.

A AS pode ser realizada em diferentes níveis, de acordo com o foco e a granularidade empregadas no processo. A abordagem mais utilizada é a organização da AS em níveis de Documento, Sentença e Aspecto [52]. Além disso, as estratégias de AS podem ser divididas quanto à técnica utilizada para solução, em processos que empregam algoritmos de Aprendizado de Máquina, estratégias baseadas em léxico e técnicas híbridas [52, 120]. Um esquema dos níveis e técnicas mais comuns utilizados na literatura é apresentado na Figura 2.1.

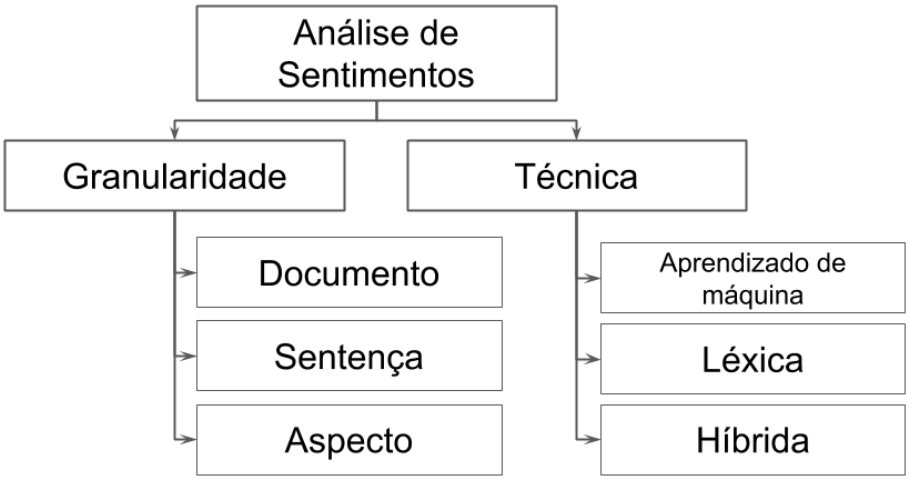

Figura 2.1: *Níveis de Análise de Sentimentos*

Na AS em nível de documento, assume-se que a entrada contém uma única opinião principal sobre um determinado assunto e não leva em consideração diferentes aspectos[1] e entidades[2] do texto. É possível, por exemplo, considerar a opinião sobre um produto como um documento e realizar a avaliação de forma geral, classificando-o como positivo, negativo ou neutro.

Considera-se, como Exemplo 1, a sentença a seguir, que ilustra uma avaliação sobre determinado produto. Em uma abordagem de AS no nível de documento, a mensagem é apreciada em sua totalidade, desconsiderando os diferentes aspectos e entidades envolvidos. Nesse caso, por exemplo, provavelmente o documento será considerado positivo pelo classificador.

Exemplo 1: "*O tablet é leve, rápido e um pouco barulhento. A bateria poderia durar mais. O brilho da tela é razoável. O sistema operacional é desatualizado. Bom custo-benefício*".

A AS no nível de sentença considera que há uma única opinião sobre determinado assunto em cada frase. Uma sentença pode até mesmo possuir sentimentos conflitantes, porém a AS nesse nível deseja saber o sentimento principal contido no período.

[1] Característica de uma entidade
[2] Objeto que é alvo da AS

Essa abordagem é semelhante a uma classificação de subjetividade de mensagens, que busca diferenciar as frases subjetivas – que contém opiniões – das frases objetivas, que expressam fatos [166]. Apesar da classificação de sentenças geralmente ocorrer em mensagens subjetivas, em [106] o autor alerta que mesmo algumas frases objetivas podem conter sentimentos. Por exemplo, a sentença do Exemplo 2 é objetiva, mas contém um sentimento negativo.

Exemplo 2: "*Comprei o tablet, mas ele já deu problemas no botão de volume*".

Além do problema anterior, outro grande desafio para a AS são as sentenças sarcásticas. Considere a frase do Exemplo 3 que faz uso de sarcarmo para expressar uma opinião negativa sobre a entidade. Apesar de não ser muito utilizada em avaliações de produtos, esse tipo de figura de linguagem é amplamente empregado em um contexto de discussão política, por exemplo [106].

Exemplo 3: "*Mas que maravilha de tablet, quebrou em 3 dias! Ótimo!*".

A análise de mensagens sarcásticas é considerada um dos grandes desafios da área de AS e vem sendo alvo de diversos estudos nos últimos anos [149, 106]. Há, inclusive, uma limitação humana no entendimento desse tipo de mensagem, que depende fortemente do conhecimento do contexto ao qual está inserida [63].

A AS em nível de aspecto leva em consideração a opinião sobre diversos tópicos de uma mesma entidade, sabendo que um documento pode possuir opiniões sobre uma série de características. Fazendo uso do Exemplo 1, uma análise nesse nível consideraria os vários aspectos da avaliação, processando o sentimento de cada um deles separadamente, conforme apresentado na frase a seguir e na Tabela 2.1.

"*O tablet é (1) leve, (2) rápido e um (3) pouco barulhento. A (4) bateria poderia durar mais. O brilho da (5) tela é razoável. O (6) sistema operacional é desatualizado. Bom (7) custo-benefício*"

Tabela 2.1: *Exemplo de Análise de Sentimentos em nível de aspecto*

Aspecto	**Opinião**	**Sentimento**
(1) Peso	"*Leve*"	Positivo
(2) Velocidade	"*Rápido*"	Positivo
(3) Ruído	"*Barulhento*"	Negativo
(4) Bateria	"*Poderia durar mais*"	Negativo
(5) Tela	"*Brilho razoável*"	Positivo
(6) Sistema Operacional	"*Desatualizado*"	Negativo
(7) Custo-benefício	"*Bom*"	Positivo

É possível perceber que a classificação da opinião inteiramente como positiva – como acontece no exemplo de AS em nível de documento – não permite identificar características negativas sobre o produto. Um comentário positivo sobre algum aspecto do objeto não significa necessariamente que o usuário gostou de todas as suas características e isso que pode ser uma limitação considerável em aplicações comerciais, por exemplo [158].

Nem sempre é possível identificar facilmente os aspectos de uma entidade, pois algumas vezes estão implícitos na opinião e devem ser descobertos para que seja feita a análise do sentimento. No Exemplo 4, a frase está tratando, implicitamente, do tamanho do objeto *tablet* [106].

Exemplo 4: "*Esse tablet não cabe facilmente na mochila*".

Dada sua importância para a AS, há uma série de trabalhos de pesquisa que se dedicam especificamente ao processo de identificação de aspectos em mensagens. O leitor interessado pode encontrar mais informações sobre o tópico em [7, 71, 193, 52].

2.1.1 Análise de Sentimentos baseada em Aprendizado de Máquina

A Aprendizagem de Máquina é uma área de pesquisa que visa o desenvolvimento de métodos capazes de deduzir conhecimento sobre determinado domínio a partir de um conjunto de dados de entrada. Essas técnicas buscam inferir automaticamente padrões nesse conjunto de forma a avaliar novos dados e classificá-los de forma correta, por meio da generalização inicial [173]. As técnicas de AM são divididas em supervisionadas e não supervisionadas.

Na AS, a abordagem mais utilizada é a supervisionada, mais especificamente a classificação [20]. Em soluções supervisionadas, técnicas de AM são aplicadas a mensagens previamente rotuladas de forma a identificar características que auxiliem na distinção e detecção de sentimentos em sentenças desconhecidas. Como a AS é essencialmente um problema de classificação de textos, qualquer método de AM pode ser utilizado [106].

Dentre as principais técnicas de AM utilizadas para a classificação de sentimentos, destacam-se o *Support Vector Machines* (SVM), o *Naïve Bayes* (NB), a Regressão logística (RL), entre outras [175]. Assim como em outras aplicações que fazem uso de AM, a escolha de boas *features*[3] é fundamental. É importante que sejam selecionadas características que permitam a diferenciação entre as opiniões que serão classificadas.

Em pesquisas de PLN e AS, geralmente as próprias palavras do texto são utilizadas como *features* [22]. Uma representação amplamente utilizada nos trabalhos

[3] Características ou atributos que serão considerados no processo de AM

é a *Bag-of-words*, que faz uso de n-gramas da mensagem original (os conceitos de *Bag-of-words* e n-gramas serão discutidos nos próximos parágrafos). Dentre outros atributos comumente utilizados, podem-se destacar: *Part of Speech* (PoS), frequência dos termos, expressões de negação, etc. Uma lista contendo as principais *features* utilizadas nos trabalhos relacionados é apresentada na Tabela 3.3.

Um esquema resumido do processo de AS, utilizando um algoritmo de Aprendizado de Máquina, é apresentado na Figura 2.2.

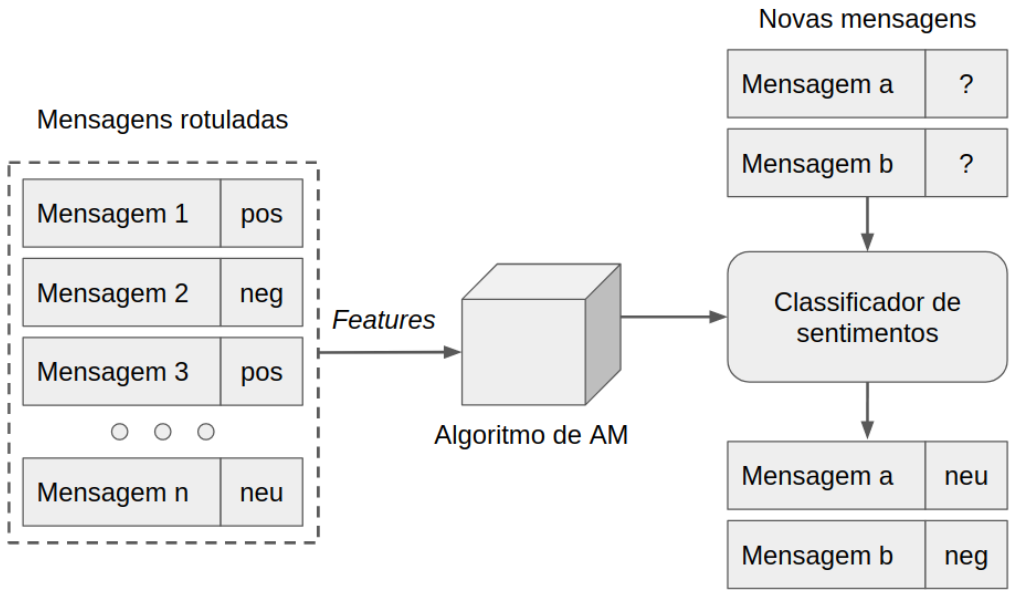

Figura 2.2: *Esquema de um processo de Análise de Sentimentos utilizando Aprendizagem de Máquina Supervisionada*

Como discutido anteriormente, a escolha dos atributos a serem considerados para o processo de criação de um classificador é um passo fundamental no projeto da solução. A seguir, será apresentada uma visão geral das principais *features* utilizadas em trabalhos de AS [2, 22, 175].

Part-of-Speech

O *Part-of-Speech* (PoS) refere-se a definição das classes gramaticais a cada qual palavra pertence, de forma a facilitar o entendimento do seu comportamento em uma frase. O *PoS Tagging*, por sua vez, consiste em processar cada termo, atribuindo seu PoS apropriado [174]. Por exemplo, a frase "*O tablet é lindo, leve, perfeito*" seria rotulada conforme a Tabela 2.2.

Tabela 2.2: *Exemplo de PoS Tagging de uma frase*

O	tablet	é	lindo	leve	perfeito
artigo	substantivo	verbo	adjetivo	adjetivo	adjetivo

O *PoS Tagging* é muito utilizado na AS, uma vez que algumas classes de palavras são boas indicadoras de sentimentos contidos em uma frase, como adjetivos e advérbios [174]. Um dos padrões mais utilizados para o *PoS Tagging* de palavras é o *Penn Treebank Tagset*, criado por [168]. A Tabela 2.3 apresenta alguns exemplos de *tags* e seus significados.

Tabela 2.3: *Tags do Penn Treebank Tagset*

Tag	Significado
CC	Conjunção coordenativa
VB	Verbo
VBD	Verbo no passado
VBG	Verbo no gerúndio
VBN	Verbo no particípio
VBP	Verbo no presente
UH	Interjeição
PRP	Pronome pessoal
PRP$	Pronome possessivo
JJ	Adjetivo
JJ	Adjetivo superlativo
SYM	Símbolo
RB	Advérbio
RBR	Advérbio comparativo
NN	Substantivo
NNS	Substantivo no plural

Frequência dos termos

A frequência com que um termo aparece em um texto pode ser relevante para a AS. Uma das técnicas mais utilizadas para o cálculo dessa medida é o TF-IDF (*Term Frequency – Inverse Document Frequency*), que identifica a importância de uma palavra em um determinado documento, baseado na quantidade de vezes em que é utilizada.

O cálculo do TF, como demonstrado na Equação 2-1, leva em consideração a frequência f de um termo t em um documento d dividido pela soma da frequência de todos os n termos do documento. O IDF calcula a quantidade total de documentos D dividido pelo número de documentos que contém o termo t, como pode ser visto na Equação 2-2. O TF-IDF é calculado pela multiplicação do TF com o IDF, conforme

Equação 2-3.

$$TF_{t,d} = \frac{f_{t,d}}{\sum_{i=1}^{n} f_{i,d}} \tag{2-1}$$

$$IDF_{t,D} = \frac{D}{\#D \subset t} \tag{2-2}$$

$$TFIDF = TF_{t,d} \times IDF_{t,D} \tag{2-3}$$

A lógica de funcionamento do TF-IDF é que as palavras que aparecem com mais frequência em um determinado documento possuem mais informações sobre a sua ideia geral, em detrimento dos termos que aparecem poucas vezes. Apesar disso, em [174] a autora discorre que, no contexto da AS, nem sempre o TF-IDF apresenta um resultado satisfatório. Em alguns casos, a simples constatação da presença de termos é mais eficaz que a identificação de sua frequência [140].

Palavras de negação

A identificação e o processamento adequados das palavras de negação exercem papel fundamental na AS, uma vez que, frequentemente, esses termos podem causar uma mudança na polaridade da mensagem. A técnica mais utilizada nos trabalhos da literatura é a inversão simples da polaridade da mensagem quando uma palavra de negação é encontrada [175].

Alguns autores, entretanto, discutem que a técnica de troca da polaridade nem sempre traz resultados corretos. Em [90], o autor mostra que, apesar da negação de palavras positivas resultar em sentimentos negativos, o contrário não é verdadeiro, ou seja, a negação de sentimentos negativos tende a manter a orientação negativa da mensagem. No mesmo estudo, foram produzidos dois dicionários léxicos, um com termos frequentes em contextos de negação e outro com termos habituais em contextos sem a presença de palavras de negação.

Por exemplo, a frase "*Esse tablet não é bom!*", em um processo de AS fazendo uso de dicionário léxico, sem considerar as palavras de inversão, provavelmente seria avaliada como positiva, por conta da palavra "*bom*". Esse termo, individualmente, representa um sentimento positivo sobre alguma entidade. O tratamento de palavras de negação é um desafio para a área e alguns trabalhos abordam exclusivamente essa questão no contexto de AS e suas implicações no resultado final do processo de análise [172].

Palavras de intensificação

Assim como acontece com as palavras de negação, uma correta identificação dos termos de intensificação pode aprimorar o poder preditivo do modelo de classificação. Essas palavras têm a capacidade de incrementar a polaridade de outras partes da frase e, com isso, podem identificar quais características possuem o sentimento mais intenso, seja ele positivo ou negativo.

Há várias formas de tratar essa tarefa no contexto de AS. A mais comum é por meio da atribuição de um fator de multiplicação para as palavras relacionadas à entrada de intensificação, aumentando as suas polaridades intrínsecas.

Por exemplo, a frase "*A tela não é resistente mas a bateria é muito boa!*" possui a palavra de intensificação "muito", que pode ser uma boa pista para um acréscimo de força do sentimento do usuário com relação ao aspecto "bateria".

Palavras de intensificação são frequentemente utilizadas como atributos em pesquisas de AS que utilizam métodos de AM [196, 87, 55], como será demonstrado no Capítulo 3, em especial na Tabela 3.3.

Bag-of-words

A técnica de *Bag-of-words* (BoW) consiste na representação da frase em um vetor de palavras, com os índices representando a frequência de determinada palavra p em uma mensagem m. Alguns autores fazem uso de uma variação com índice binário, indicando somente a presença (valor 1) de uma palavra p em uma mensagem m ou a sua ausência (valor zero) [175]. Por exemplo, a frase "*Adorei esse tablet, adorei a tela, adorei a performance.*" pode ser representada em BoW, usando a estratégia de frequência de palavras, como demontrado na Tabela 2.4.

Tabela 2.4: *Exemplo de representação de uma frase por meio da técnica Bag-of-words*

Palavra	Adorei	esse	tablet	a	tela	performance
Frequência	3	1	1	2	1	1

N-gramas

Na análise textual para a AS, pode-se levar em consideração palavras únicas ou a sequência de termos. Um n-grama é uma série contínua de n palavras de um texto. As representações mais utilizadas nas pesquisas de AS apresentam os valores de $n = 1$ (unigrama), $n = 2$ (bigrama) e, em alguns casos, $n = 3$ (trigramas). O uso dessas variações justifica-se pelo fato de que, muitas vezes, é necessário analisar mais de uma palavra para poder avaliar corretamente o significado da mensagem [154, 161]. Por exemplo,

considerando a frase "*Esse tablet não é bom*", as representações em unigramas, bigramas e trigramas são apresentadas na Figura 2.3.

Figura 2.3: *Representação de uma frase em unigrama, bigrama e trigrama*

2.1.2 Análise de Sentimentos baseada em Dicionários Léxicos

Dicionários léxicos podem ser definidos como um conjunto de palavras e suas respectivas polaridades – grau de positividade e negatividade de um termo. Frequentemente, esses valores são expressos por meio de dados numéricos que representam a intensidade do sentimento. Técnicas baseadas em léxicos atuam principalmente em características sintáticas e semânticas do texto e fazem uso de dicionários léxicos. A partir desse dicionário, é feito o processamento das mensagens pelo classificador e retornada a polaridade de cada uma delas [15].

O uso de dicionários léxicos apresenta uma série de vantagens. O conteúdo linguístico pode ser levado em consideração utilizando, por exemplo, palavras de negação e intensificação. Além disso, as orientações semânticas das entidades podem diferenciar-se de acordo com suas particularidades. Algumas características dependentes de uma linguagem específica também podem ser incluídas nessas abordagens [13].

Apesar disso, algumas desvantagens da utilização da abordagem léxica devem ser consideradas em uma estratégia de AS. A principal delas é a dificuldade de criação e manutenção de um dicionário léxico confiável e consistente, que é um dos desafios para pesquisas que fazem uso dessa técnica. Além disso, outro aspecto importante é a dependência de um domínio específico para o qual o léxico foi construído, bem como o idioma no qual foi escrito.

Essas características tornam difícil a tarefa de manter um dicionário léxico de uso geral, independente de domínio [13, 150]. Algumas palavras possuem orientação semântica diferente, dependendo do domínio em que estão inseridas e até mesmo da função sintática que exercem na frase. A palavra "*câncer*", por exemplo, pode não possuir um significado negativo em um contexto técnico, diferentemente de um contexto geral.

Frequentemente, trabalhos de pesquisa na área de AS utilizam mais de um dicionário para apoiar o processo de classificação. A Tabela 3.2, apresentada no Capítulo

3, demonstra os léxicos empregados por alguns trabalhos na área de AS e que serviram como base para a pesquisa.

Em [11], o autor analisa uma série de trabalhos de AS e atesta que, via de regra, as técnicas com melhores resultados gerais possuem em comum a característica de utilização de um conjunto de dicionários léxicos. O autor confirma a hipótese de que a combinação desses dicionários implica em melhores resultados na classificação de mensagens. Alguns dicionários, entretanto, são gerados de forma manual e possuem problemas de escalabilidade e dependência de domínio, semelhantes aos enfrentados pelos classificadores criados manualmente [47].

Considerando o exposto, é possível notar que uma das principais limitações da utilização do dicionário léxico é a própria lista de palavras disponíveis. Esse fato, muitas vezes, limita a realização de uma AS mais profunda e com melhores resultados em determinado contexto [47]. Nesse sentido, outro desafio da área é a criação e expansão de léxicos, com diversas pesquisas realizadas especificamente com essa finalidade [195, 58, 86].

Existem, basicamente, 3 formas de criação e expansão de um dicionário léxico: manual – processo realizado por especialistas humanos que analisam cada palavra, atribuindo uma polaridade para cada uma delas – e duas formas (semi) automatizadas: baseada em dicionário e baseada em corpus.

A técnica de expansão léxica baseada em dicionários faz uso de um processo de busca por sinônimos e antônimos de termos com polaridade conhecida e consolidada – chamadas de palavras semente, *seed words* em inglês – de forma a expandir o conjunto de entradas. O mais famoso dicionário de sinônimos e antônimos utilizado na literatura é o *Wordnet*[4]. A técnica baseada em corpus emprega um conjunto de documentos de um determinado domínio de forma a descobrir e atribuir polaridade para cada uma das palavras desse contexto. Frequentemente, essas técnicas são utilizadas em conjunto, principalmente a validação manual de dicionários criados de forma automatizada. O conjunto de léxicos utilizados nesta pesquisa é apresentado no Capítulo 4, mais precisamente na Tabela 4.4.

2.1.3 Análise de Sentimentos baseada em abordagem híbrida

Abordagens híbridas utilizam tanto técnicas de AM quanto baseadas em léxico para a classificação das mensagens e geralmente os dicionários têm papel central nesse processo [120]. Em [2], o autor faz uma análise dos principais atributos utilizados em classificadores de sentimentos e conclui que o uso das polaridades de palavras

[4] `https://wordnet.princeton.edu/`

disponibilizadas por dicionários léxicos é uma das características mais importantes para incrementar o poder de predição dos métodos de classificação.

2.1.4 Análise de Sentimentos em *Tweets*

A utilização de mensagens do *Twitter* para a AS é muito comum, sendo inclusive considerada uma área de pesquisa especializada da AS tradicional, a Análise de Sentimentos de *Tweets* (AST) [197]. O uso do *Twitter* como uma das principais fontes de pesquisa na área de AS justifica-se, principalmente, por sua consolidação como uma plataforma global de geração e consumo de conteúdo.

Dados do último ano mostram que, em média, 500 milhões de *tweets* são gerados por dia no site, 80% deles escritos por meio de um dispositivo móvel [137]. Além disso, aproximadamente 350 milhões de pessoas acessam o site ativamente, o que reflete a grandeza e a importância da ferramenta [137].

Por conta da característica dos *tweets*, a AST apresenta uma série de desafios e alguns pontos importantes das mensagens devem ser considerados, como a limitação do tamanho do texto em 280 caracteres, a grande quantidade de dados gerados, o estilo informal de escrita, a variedade de tópicos de discussão, entre outros [175].

Por se tratar de uma rede social, muitas mensagens são escritas utilizando linguagem informal, abreviações, gírias e afins. Isso pode dificultar a AST, principalmente em técnicas que usam dicionários léxicos, uma vez que as palavras não são armazenadas em todas as suas variações, o que muitas vezes demanda alguns procedimentos de pré-processamento e normalização das mensagens.

A limitação no tamanho das mensagens faz com que os usuários utilizem construções mais diretas, com o uso de poucas palavras. Esse é outro desafio para técnicas baseadas em léxico, que podem não possuir os termos utilizados no *tweet*. A maioria das mensagens possui apenas uma sentença, por isso a maior parte dos trabalhos de AST abordam o problema nessa granularidade [175].

O estilo de linguagem e o contexto do idioma também são desafios para trabalhos na área. Há uma grande variação na forma com que as mensagens são escritas na rede, dependendo de diversos fatores, como o usuário que redigiu o *tweet*, a idade, o tema, o público-alvo, entre outros. Também, por se tratar de uma rede social amplamente utilizada, há diferenças culturais e de idioma nos *tweets*, tornando a AST ainda mais complexa.

O *Twitter* também possui alguns termos específicos, utilizados no contexto das mensagens dos usuários. Uma *hashtag* indica um tópico na rede e são precedidos pelo caracter #. Usuários podem fazer menção a outros em suas mensagens, por meio do caracter @ seguido do pseudônimo da pessoa desejada. Um *retweet* é o processo de

propagação de um *tweet* original por um usuário da rede [175]. Um exemplo de *tweet* do usuário *@airtonbjunior* que possui *hashtag* (#ml) e menção ao usuário @coursera é apresentado na Figura 2.4. Pesquisas na área de AST são apresentadas na Seção 3.1.

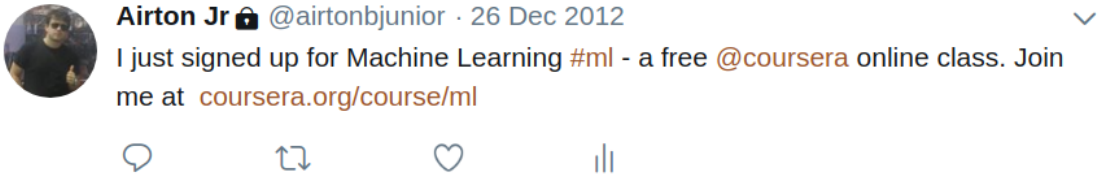

Figura 2.4: *Exemplo de tweet*

2.1.5 Aplicações da Análise de Sentimentos

A AS pode ser aplicada em diversas áreas e apresenta-se como uma linha de pesquisa promissora, com um número crescente de trabalhos nos últimos anos. Muitos estudos têm como foco a análise de mensagens em redes sociais, com destaque para o *Twitter* [15, 17, 21, 137, 175]. Algumas aplicações permitem que o usuário acompanhe o sentimento geral sobre determinado tópico em tempo real [88].

Empresas podem fazer uso da AS para identificar as opiniões das pessoas sobre determinado produto ou marca, de forma a planejar melhorias e campanhas de *marketing* [29, 152, 151]. Alguns trabalhos utilizam a AS para a previsão de comportamento do mercado financeiro, buscando otimizar os investimentos e disparar gatilhos para auxiliar na tomada de decisões dos investidores [23, 122, 138].

Pesquisas recentes procuram realizar a predição do resultado de eleições com base na opinião geral das pessoas sobre os candidatos e suas respectivas campanhas, principalmente nas redes sociais [32, 187, 167]. Outros tópicos em que a AS pode ser aplicada incluem estimativas de receitas de filmes por meio de opiniões, previsões de vendas de determinado produto, identificação de influenciadores digitais, entre outros [106].

Governos também têm utilizado a AS para a criação de plataformas para o monitoramento de comunicações, com foco principalmente em mensagens e sentimentos hostis e negativos [174]. Há uma série de trabalhos na literatura que se dedicam a construir uma revisão sistemática das pesquisas na área de AS, identificando as principais técnicas utilizadas e o contexto em que foram aplicadas. Como sugestão ao leitor interessado, pode-se destacar os trabalhos publicados em [52, 80, 156, 120].

2.1.6 Aplicação da Análise de Sentimentos neste trabalho

O presente trabalho utiliza uma abordagem de Análise de Sentimentos em nível de documento, ou seja, analisa a mensagem de forma integral e assume que ela exprime

uma única opinião sobre determinado assunto sem considerar os seus diferentes aspectos. Esse é um comportamento comum em AST devido ao fato da limitação no tamanho dos *tweets* fazer com que os usuários utilizem construções mais diretas. Além disso, esta pesquisa usa uma estratégia híbrida para o processo de classificação das mensagens em positiva, negativa ou neutra, empregando dicionários léxicos e técnicas de Aprendizado de Máquina, por meio da Programação Genética e de outros algoritmos clássicos de AM.

2.2 Programação Genética

A Programação Genética (PG) é um campo da computação evolucionária que busca resolver problemas, de forma automatizada, sem demandar conhecimento detalhado sobre a solução [93]. De forma geral, as técnicas evolucionárias bucam soluções para problemas de otimização com base em conceitos definidos na teoria da evolução, escrita por *Charles Darwin*, como a existência de uma população de indivíduos que competem entre si, a adaptação desses ao contexo, processos de modificação, reprodução e inovação, entre outros [81, 93, 145].

De modo geral, é possível definir a PG como um método sistemático, não dependente de um domínio específico, usado para permitir que computadores criem programas para solução de problemas de forma automática, inciando com um conhecimento de alto nível sobre as regras gerais dos possíveis modelos. Nesse contexto, programa significa um modelo capaz de, a partir de uma ou mais entradas, produzir uma saída para as mesmas. Embora esses programas possam ser representados por diversos tipos de estruturas, a forma mais comum é a representação por meio de árvores. Dentre as outras possíveis estruturas usadas em PG, podem-se citar os grafos e a representação linear [116].

Pesquisas utilizando a PG vem aumentando nos últimos anos, devido ao seu sucesso em resolver problemas do mundo real. A técnica também é conhecida por obter resultados competitivos com soluções encontradas por humanos, algumas vezes até mesmo obtendo melhor desempenho em uma grande classe de problemas [145, 66]. Há, inclusive, uma lista com 36 invenções recriadas pelo processo de Programação Genética[5], 2 delas patenteadas com sucesso [94].

Ao utilizar a PG para a resolução de problemas, algumas decisões importantes devem ser tomadas. Alguns autores definem essa etapa como os passos preparatórios para a utilização da PG [93, 145]:

- Definição do conjunto de terminais;
- Definição do conjunto de funções;

[5] `http://www.genetic-programming.com/inventionmachine.html`

- Definição da função de aptidão (*fitness*);
- Definição de parâmetros de controle do processamento do algoritmo;
- Definição do critério de parada e escolha do resultado do processo.

Assim como a natureza, a PG é um processo estocástico, e não garante o resultado ótimo. Porém, essa aleatoriedade faz com que, frequentemente, as soluções fujam de problemas enfrentados por métodos determinísticos gulosos, como máximos e mínimos locais [145]. Um fluxo geral do funcionamento de um algoritmo de PG pode ser visto na Figura 2.5 e um pseudocódigo do processo é apresentado no Algoritmo 1. Os conceitos de Operadores Genéticos, Seleção, Função de Aptidão, entre outros serão explicados nas próximas seções.

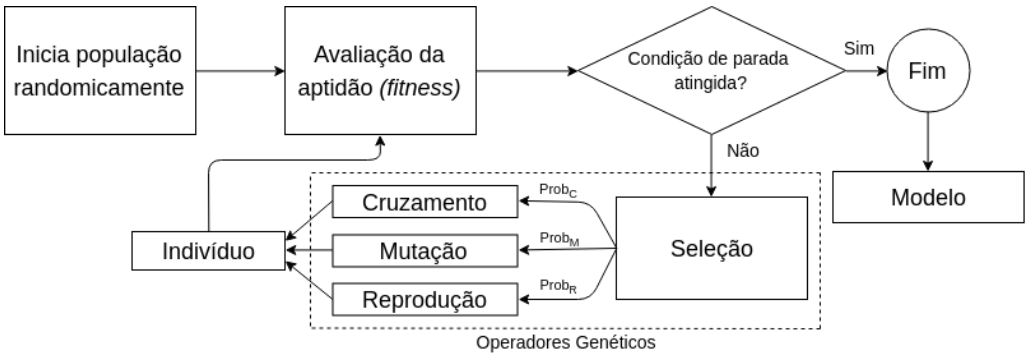

Figura 2.5: *Fluxo geral de funcionamento da Programação Genética*

Algorithm 1 Fluxo geral da Programação Genética

```
1: INICIALIZA população randomicamente
2: AVALIA aptidão dos indivíduos da população
3: while NOT condição de término do
4:     SELECIONA indivíduo com base em sua função de aptidão
5:     APLICA operadores genéticos
6:     GERA novos indivíduos
7: end while
8: return MELHOR indivíduo
```

2.2.1 Terminais e funções

Apesar da descrição inicial definir a PG como uma forma de criar programas de forma automatizada, esses programas geralmente não possuem um formato comum, utilizado na maior parte das linguagens de programação. São construídos de forma mais restrita e, frequentemente, com detalhes inerentes ao domínio específico do problema. Com isso, uma etapa inicial importante é a definição das primitivas, composta pelos

terminais e funções da PG, que serão utilizadas para a criação dos indivíduos da solução [145].

É possível, portanto, considerar que as primitivas P são compostas de um conjunto de terminais $t \in T$ e de um conjunto de funções $f \in F$ que representam o domínio do problema. Os terminais podem, ainda, ser divididos em 3 conjuntos: as entradas do programa, as funções que não possuem parâmetros e as constantes.

As entradas do programa são os argumentos passados para os indivíduos que representam os possíveis modelos da solução e geralmente são representadas por variáveis nomeadas como a, b, x, y, entre outros. As funções sem argumentos possuem aridade[6] zero. Constantes são valores predefinidos, geralmente gerados de forma aleatória.

As funções estão fortemente ligadas ao domínio do problema a ser solucionado. No caso deste trabalho, por exemplo, o problema de manipulação de linguagem natural para a análise de sentimentos demanda métodos de processamento de texto. Funções matemáticas simples, como soma, subtração, multiplicação e divisão são comumente utilizadas na maior parte dos problemas, além de funções lógicas como AND e OR.

Ao definir a coleção de primitivas para a PG, alguns aspectos relevantes devem ser considerados. Duas propriedades importantes desse conjunto são a suficiência e o fechamento.

A suficiência é a propriedade que garante que é possível chegar a uma solução válida para o problema que pretende-se resolver utilizando as primitivas P definidas nos conjuntos T de terminais e F de funções [144]. A definição insuficiente de P implica diretamente na incapacidade da técnica em solucionar o problema proposto e, no melhor caso, permite atingir apenas uma aproximação do resultado desejado. Entretanto, na prática, frequentemente não se sabe qual é o conjunto suficiente, e testes empíricos com as primitivas tornam-se necessários. Além disso, algumas vezes a aproximação do resultado é suficiente para o objetivo proposto [144].

A propriedade de fechamento determina que qualquer subárvore pode ser utilizada como entrada de qualquer função f do conjunto de funções F [144, 93]. Desse modo, são necessários processos de verificação adicionais em funções básicas como, por exemplo, a divisão, pois, como se sabe, não é possível dividir um número por zero. Em [93], o autor sugere, nesse caso, uma divisão protegida, que retorna o valor 1 caso um número tente ser dividido por zero. A formalização da propriedade de fechamento é apresentada em 2-4, sendo F' o conjunto formado por todos os resultados possíveis das funções $f \in F$.

$$\forall x \in F' \cup T, \forall f \in F, x \in \mathrm{dom}(f) \tag{2-4}$$

[6] Quantidade de parâmetros passados para uma função

Muitas vezes as soluções demandam a manipulação de diferentes tipos de dados – como é o caso deste trabalho, que processa texto (*string*), números reais (*float*) e valores lógicos (*boolean*), como será apresentado em detalhes no Capítulo 4. Nessas situações, não se pode garantir que a saída de qualquer função $f \in F$ possa ser utilizada como entrada de outra, ou seja, não é possível atender plenamente a propriedade de fechamento. Esse tipo de restrição pode ser solucionada utilizando a abordagem de Programação Genética Fortemente Tipada (PGFT), criada por [126].

Na PGFT, durante o processo de manipulação dos indivíduos, são verificadas restrições de tipo para as funções, de forma que somente entradas corretas sejam passadas como parâmetro para outras funções. Nesse caso, há um controle adicional nos processos de transformação das árvores, verificando se a saída de um nó pode ser atribuído como entrada de outro [126].

Um exemplo de um programa gerado pela PG representado por uma árvore, nas quais os nós internos representam funções e os nós folha representam terminais do problema, pode ser visto na Figura 2.6, que representa o código demonstrado no Algoritmo 2.

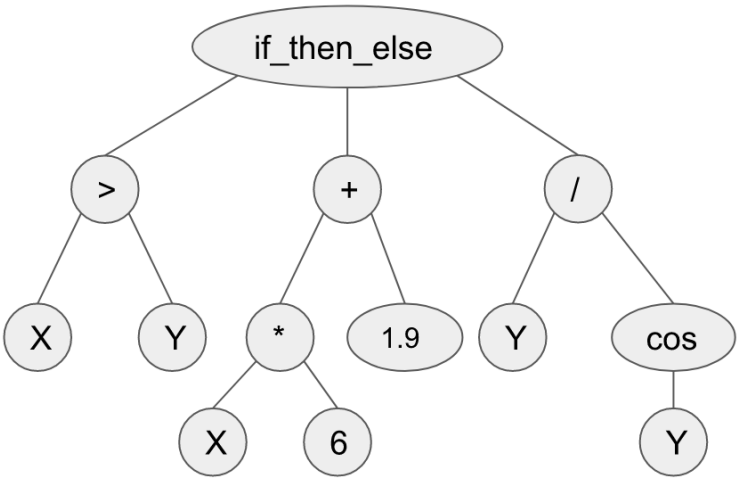

Figura 2.6: *Exemplo de um programa em Programação Genética*

Algorithm 2 Algoritmo gerado pela representação da árvore da PG

```
if X > Y then
    return X × 6 + 1.9
else
    return Y / cos(Y)
end if
```

O conjunto de terminais e funções utilizadas neste trabalho é apresentado no Capítulo 4, mais precisamente na Seção 4.1.

2.2.2 Função de aptidão

Uma das características mais importantes da Programação Genética é a função de aptidão, ou função *fitness*. Essa função busca representar, de forma quantitativa, a similaridade de cada indivíduo em relação ao resultado esperado.

A avaliação do *fitness* dos indivíduos é uma forma de identificar o nível de aptidão de cada um deles, apoiando o processo evolutivo e facilitando a escolha dos melhores para a aplicação dos operadores genéticos [157].

De forma geral, em problemas de otimização, a função de aptidão frequentemente é definida pela função objetivo do problema. Geralmente, de posse de uma base anotada de treinamento, é realizada a comparação dos valores obtidos com os resultados esperados. Indivíduos que obtiverem respostas mais próximas da correta possuem melhores valores de *fitness* [157, 108].

Assim como os outros parâmetros da PG, a escolha de uma boa função *fitness* está intimamente ligada ao tipo do problema. Aplicar uma função de aptidão que corresponde ao domínio desejado é essencial para guiar a evolução em direção à solução pretendida [157].

Diferentemente de outras técnicas de classificação, a avaliação da função *fitness* da PG demanda a execução de todos os indivíduos da população, uma vez que são formados por programas. Esse processo, na maior parte dos casos, é realizado múltiplas vezes, para que possam ser avaliados os valores de aptidão de cada um dos indivíduos e quantificar quão próximo estão do resultado desejado [145]. A função de aptidão utilizada neste trabalho é explanada em detalhes no Capítulo 4.

2.2.3 Parâmetros de controle

A definição de parâmetros de controle é uma fase essencial da configuração do algoritmo de PG. Não há uma regra geral para essa etapa e os valores definidos dependem do domínio do problema e o contexto ao qual está sendo aplicado. Apesar disso, alguns autores definem parâmetros de referência para algumas classes de problemas que já foram amplamente exploradas por meio da PG [145].

Dentre os principais parâmetros da PG é possível destacar o tamanho da população e a quantidade de gerações de indivíduos [93, 145]. Em [111], o autor afirma que a configuração mais comum é a utilização de uma quantidade pequena de gerações, usualmente 50, e população variando entre 500 e 2000. A razão para esses valores é, principalmente, cultural, uma vez que foram as configurações definidas por *John Koza* [7] em

[7] Cientista da Computação, conhecido como o pioneiro no uso da Programação Genética para resolução de problemas complexos

[93].

Além disso, alguns autores afirmam que poucas evoluções ocorrem nos indivíduos após a 51ª geração [145]. Apesar disso, para o presente trabalho, testes empíricos demonstraram que o aumento no número de gerações resultaram em modelos com maior capacidade de predição, como apresentado no Capítulo 5.

A definição das probabilidades para cada um dos operadores genéticos (discutidos na Seção 2.2.6) também é feita nessa etapa. A taxa de escolha de indivíduos para cruzamento costuma ser alta, frequentemente acima de 90% [145, 116]. A mutação, considerada como uma operação genética secundária por [93], geralmente possui um valor baixo, entre 1 e 5%.

Apesar de [93] não considerar necessário o operador de mutação, com o argumento de que a PG não é uma busca puramente aleatória, outras pesquisas mostram que os processos evolutivos podem beneficiar-se desse recurso de modificação [116]. O operador de seleção costuma ocorrer quando a soma da probabilidade de cruzamento e mutação é menor que 100%, tendo como probabilidade definida por $1 - p$, sendo $p = prob_{cruzamento} + prob_{mutacao}$.

A definição da profundidade máxima da árvore também é um critério importante e que influencia diretamente na criação da população inicial (Seção 2.2.5) e nos indivíduos de cada geração. Uma apresentação detalhada dos parâmetros de controle possíveis utilizados em Programação Genética pode ser vista em [93].

2.2.4 Critério de parada e solução

As formas mais comuns de determinar que um algoritmo de PG deve ser finalizado é por meio da definição de um número máximo de gerações e a indicação de um erro aceitável para a solução encontrada. Variações desses métodos também podem ser aplicadas, como uma quantidade máxima de gerações sem evoluções do melhor indivíduo.

Na maior parte das vezes, a solução do problema é representada pelo melhor indivíduo avaliado entre todas as gerações. Porém, em alguns casos, um conjunto de indivíduos pode ser retornado, dependendo do problema em que a PG está sendo aplicada [145].

2.2.5 População inicial

Na PG, assim como em outros algoritmos baseados na teoria da evolução, são criadas populações em que cada indivíduo representa uma possível solução para o problema. A inicialização aleatória é a forma mais comum de criação da população, que evolui no decorrer dos ciclos, chamados de gerações. A cada geração, indivíduos possivelmente melhores são criados, evoluindo os programas (modelos) gerados.

A criação de um indivíduo na PG começa com a escolha de uma função $f \in F$ para compor a raiz da árvore. Na sequência, os argumentos da raiz são selecionados de forma recursiva até que a árvore seja finalizada com os nós representados por elementos t pertencentes ao conjunto de terminais T [116].

O conjunto das possíveis estruturas em Programação Genética é formado por todas as possíveis combinações de funções que são compostas recursivamente por elementos do conjunto $F = \{f_1, f_2, ..., f_n\}$ representando as n funções do problema e do conjunto de m terminais $T = \{t_1, t_2, ..., t_m\}$. Cada função $f^Z \in F$ possui uma quantidade Z de argumentos, representando sua aridade [93].

Existem diferentes técnicas para a criação da população na PG. As principais formas são a *Grow*, *Full* e *Ramped half-and-half* [93]. Uma propriedade importante para esse processo é a altura máxima da árvore A, ou seja, o maior caminho da raíz até um nó folha.

Na técnica *Full*, a população inicial é gerada com todos os indivíduos possuindo a altura máxima definida tendo, portanto, todas as folhas na mesma profundidade. Ou seja, sendo A o atributo que representa essa altura máxima, esse método escolhe aleatoriamente funções $f \in F$ até uma profundidade $A-1$ e, então, faz a escolha somente de terminais $t \in T$ [67]. A Figura 2.7 apresenta uma população de 2 indivíduos criados com a técnica *full* e altura máxima $A = 2$.

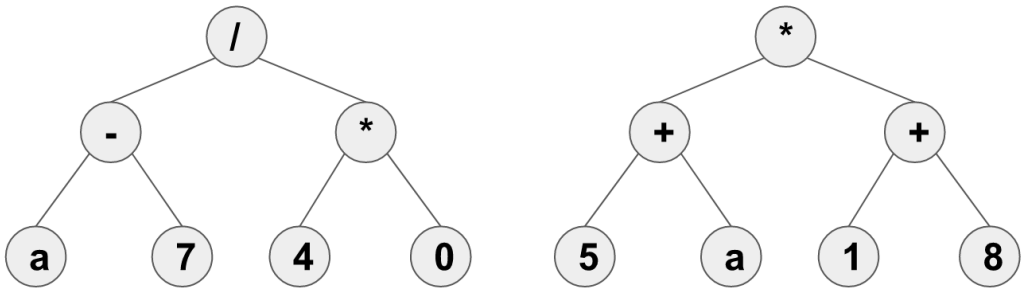

Figura 2.7: *Exemplo de população inicial de 2 indivíduos criada com o método Full, utilizando como parâmetro de altura máxima 2 níveis*

No modo *Grow*, a população é criada de forma aleatória, com profundidade variável, respeitando a profundidade máxima definida pelo parâmetro A. Essa estratégia cria árvores irregulares. Um determinado ramo termina quando atinge a altura máxima A ou quando um terminal $t \in T$ é escolhido [67]. Dois indivíduos criados com a técnica *Grow* utilizando $A = 2$ podem ser vistos na Figura 2.8.

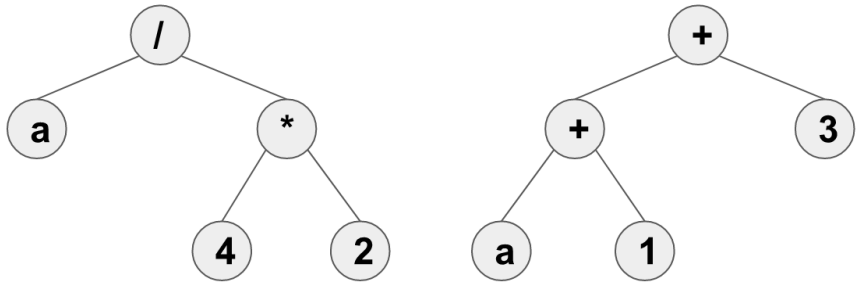

Figura 2.8: *Exemplo de população inicial de 2 indivíduos criada com o método Grow, utilizando como parâmetro de altura máxima 2 níveis*

O método *Ramped half-and-half* é o mais utilizado em pesquisas utilizando PG e gera 50% da população por meio do método *Full* e 50% fazendo uso do método *Grow*. O objetivo principal dessa estrtatégia é garantir uma população variável em termos de tamanho e forma [67]. A Figura 2.9 mostra uma população de dois indivíduos criados com essa técnica.

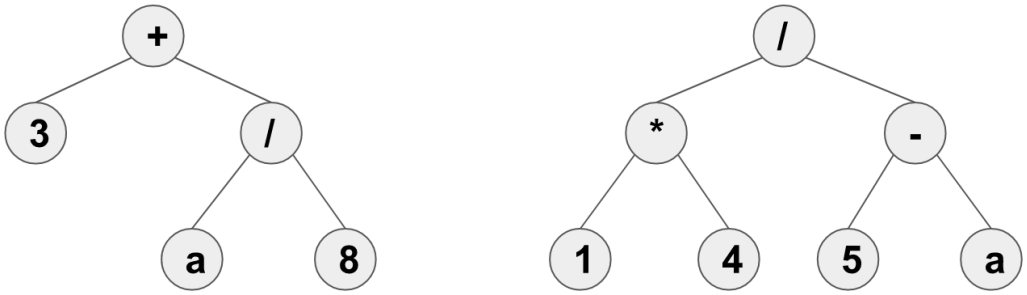

Figura 2.9: *Exemplo de população inicial de 2 indivíduos criada com o método Ramped half-and-half, utilizando como parâmetro de altura máxima 2 níveis*

Métodos alternativos para a criação de indivíduos na Programação Genética são discutidos em alguns trabalhos na literatura, como [116, 145, 112].

2.2.6 Operadores Genéticos

Os responsáveis pela evolução da população de indivíduos são os operadores genéticos. Para a Programação Genética, os principais operadores são a reprodução, mutação e cruzamento (*crossover*). Além desses principais operadores, existem outros como a edição, encapsulamento e a destruição [142].

Em PG, geralmente o indivíduo é submetido a apenas um dos operadores genéticos de forma excludente, ou seja, caso seja escolhido para participar do cruzamento, por exemplo, não será selecionado para a mutação.

Um indivíduo é selecionado para ser processado por um dos operadores genéticos baseado em sua função de aptidão, ou seja, indivíduos com melhor *fitness* tem mais

chances de replicarem seu código para futuras gerações. As principais formas de seleção de indivíduos são a Roleta e o Torneio.

Na Seleção por Roleta, cada indivíduo é representado em uma roleta virtual por uma porção proporcional ao seu *fitness*, ou seja, indivíduos com maior aptidão recebem uma fatia maior e, portanto, têm mais chances de serem selecionados. Consequentemente, os indivíduos menos aptos, apesar de ainda poderem ser escolhidos, possuem menos chances de passarem seus códigos para as próximas gerações. Em termos práticos, é uma seleção feita de forma aleatória, porém influenciada diretamente pelas aptidões dos participantes [116].

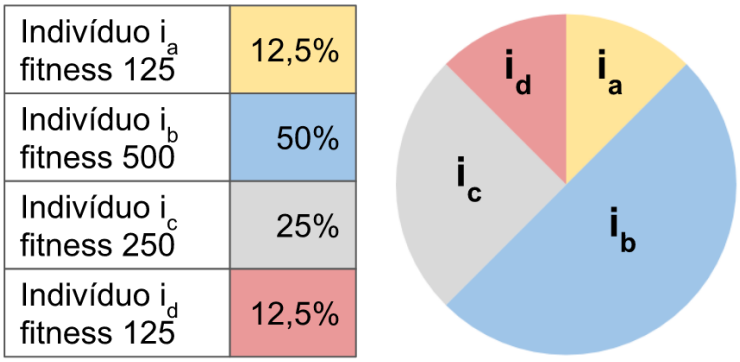

Figura 2.10: *Exemplo de Seleção por Roleta*

O cálculo da probabilidade de um indivíduo i ser escolhido na Seleção por Roleta pode ser vista na Equação 2-5, na qual i representa o i-ésimo indivíduo da população de N indivíduos e f_i representa seu *fitness*. Uma ilustração do funcionamento geral da Seleção por Roleta é apresentada na Figura 2.10.

$$prob_i = \frac{f_i}{\sum_{i=1}^{N} f_i} \tag{2-5}$$

Na Seleção por Torneio, são escolhidos k indivíduos de forma aleatória, e o que possui maior aptidão entre eles é selecionado para passar pelos operadores genéticos. É o método de seleção mais utilizado por algoritmos de Programação Genética [116]. Essa abordagem tem como principal benefício impedir que um indivíduo com uma função de aptidão muito superior aos outros domine a escolha, uma vez que a seleção para o torneio é feita de forma aleatória e a competição acontece apenas entre os k indivíduos escolhidos em cada rodada.

A Figura 2.11 apresenta um exemplo de Seleção por Torneio, utilizando $k = 3$. Perceba que, nessa situação, o indivíduo i_1 foi selecionado no torneio (4), mas por conta de seu baixo valor de aptidão (*fitness*) teria poucas chances de ser escolhido caso fosse utilizada a técnica de Roleta.

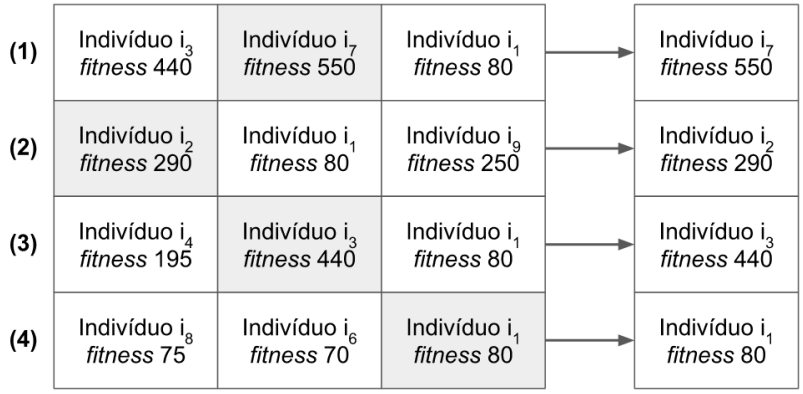

Figura 2.11: *Exemplo de Seleção por Torneio utilizando k=3*

Reprodução

Na reprodução, também chamada de seleção direta, um indivíduo é escolhido para fazer parte da próxima geração sem nenhuma modificação. Grande parte da demanda de processamento da PG está na avaliação da aptidão do indivíduo. Em termos computacionais, portanto, a reprodução é o operador genético mais performático, uma vez que faz uma cópia simples do modelo para a próxima geração [116].

Cruzamento

O cruzamento (*crossover*) é um operador que faz a recombinação do material genético de indivíduos selecionados. Em sua forma mais comum, um nó aleatório é escolhido em cada um dos participantes do *crossover* e é feita uma permuta das subárvores que tem como raíz os nós escolhidos. Ao selecionar dois pais para o processo de cruzamento, portanto, dois filhos são gerados, conforme pode ser observado na Figura 2.12, em que os pais a e b geram os filhos c e d.

Em [67] o autor disserta que o cruzamento é um operador que pode promover grandes saltos no espaço de busca ao mesmo tempo em que pode ser uma operação destrutiva. Isso se deve ao fato de que, por conta da aleatoriedade na escolha dos nós permutantes, blocos de código com boa aptidão – também chamados de blocos construtores [99, 81] – podem ser quebrados, podendo diminuir consideravalmente o *fitness* do novo indivíduo gerado. Esse fenômeno – quando os filhos apresentam funções de aptidão piores que seus pais – é conhecido como cruzamento destrutivo (*destructive crossover*) [81]. Métodos alternativos e maiores discussões sobre o processo de cruzamento podem ser lidos em [81, 93, 145].

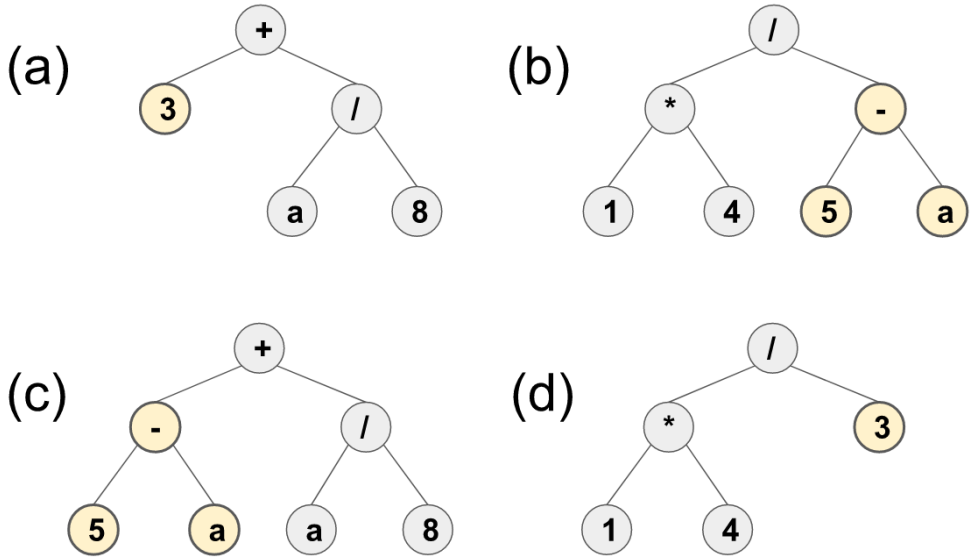

Figura 2.12: *Exemplo de cruzamento de pais (a) e (b) gerando os filhos (c) e (d)*

Mutação

O operador de mutação é uma importante ferramenta para prover maior diversidade para a população de indivíduos e manter a exploração do espaço de busca. Essa operação altera um trecho da árvore, de forma a criar um indivíduo modificado em pontos escolhidos aleatoriamente. As duas principais formas de mutação são a mutação pontual e a mutação macro [67].

A mutação pontual escolhe um nó da árvore de forma aleatória e o substitui por um correspondente possível do mesmo conjunto. Essa operação deve levar em consideração a categoria do nó (função ou terminal) e a aridade do elemento, no caso de funções [67]. Na utilização de PGFT, a verificação adicional do tipo do nó deve ser realizada. Um exemplo de mutação pontual em um indivíduo *a* gerando um indivíduo *b* pode ser vista na Figura 2.13.

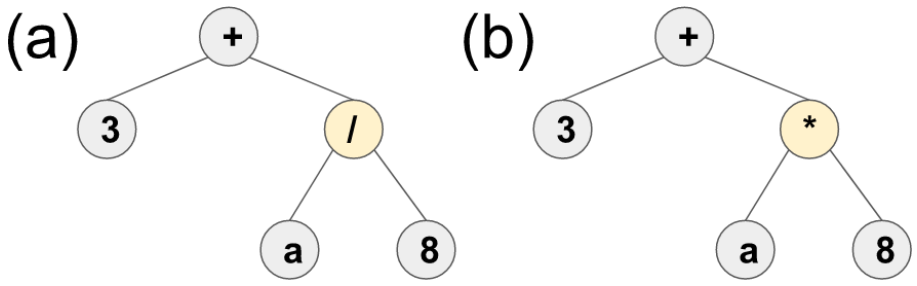

Figura 2.13: *Exemplo de mutação pontual de indivíduo (a) gerando indivíduo (b)*

Na mutação macro, um nó da árvore – escolhido de forma aleatória – é removido juntamente com seus nós adjacentes. Em seu lugar é inserido uma nova árvore criada

pelo mesmo processo de geração de indivíduos da população inicial da PG [67]. A Figura 2.14 mostra um exemplo de mutação macro em um indivíduo *a* gerando um indivíduo *b* modificado.

O operador de mutação foi considerado desnecessário em [93], com o argumento central de que a PG não é uma busca puramente aleatória de pontos no espaço de soluções. O autor argumenta que a característica heterogênea de tamanhos e formas das árvores já provê a propriedade de diversidade na população. Pesquisas posteriores mostraram, entretanto, que a mutação pode auxiliar no processo de evolução dos indivíduos, contribuindo com a resolução do problema [144, 116].

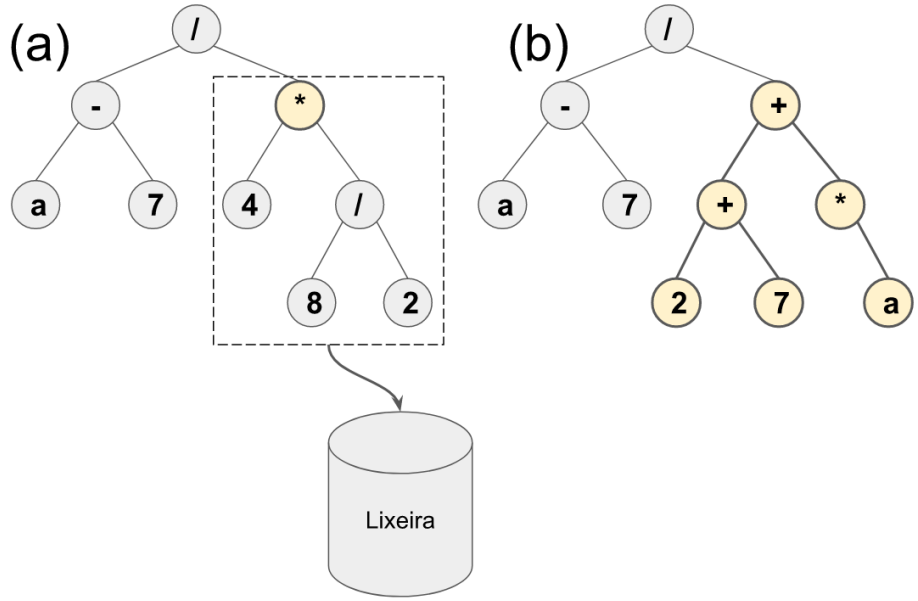

Figura 2.14: *Exemplo de mutação macro de indivíduo (a) gerando indivíduo (b)*

Elitismo

Como pode-se observar, um bom *fitness* não garante que um indivíduo seja selecionado para ser submetido aos operadores genéticos e, consequentemente, para as próximas gerações. Algumas vezes, é interessante que esses indivíduos sejam automaticamente copiados para as gerações seguintes, de forma a preservar sua estrutura. O Elitismo é o mecanismo que garante que isso seja realizado.

Em uma população de N indivíduos, são selecionados os M melhores para que sejam copiados automaticamente para a geração seguinte [67]. Note que o elitismo é diferente do operador genético de reprodução (Seção 2.2.6), uma vez que, em uma PG que usa uma configuração de elitista de M indivíduos, a cada ciclo os M melhores serão copiados para a próxima geração, independente de probabilidade [81].

2.3 Combinação de Classificadores

A maior parte das soluções de Análise de Sentimentos por meio de Aprendizado de Máquina utiliza apenas um classificador [148, 129, 18, 175]. É comum, inclusive, que várias técnicas sejam empregadas, com o objetivo de encontrar uma que maximize a quantidade de acertos e, então, utilizá-la para a classificação [37].

Ao escolher somente o melhor classificador entre todos os gerados, informações importantes sobre os dados podem ser perdidas [175]. Uma forma de aproveitar o conhecimento obtido por todas as técnicas sobre os elementos analisados é por meio da combinação desses classificadores, técnica conhecida como Comitês de Classificadores ou *ensemble*.

Os *ensembles* são sistemas compostos por um conjunto de classificadores – também chamados de classificadores de base – e por um método que realiza a combinação dos resultados de cada um deles [38]. Esses comitês buscam aproveitar a variedade dos modelos gerados, combinando-os com o objetivo de encontrar melhores soluções para o problema [175].

A melhoria nos resultados das predições dos comitês estão diretamente relacionadas com o desempenho individual de cada um dos modelos utilizados, ou seja, nos casos em que todos os classificadores são idênticos, o *ensemble* pode ser substituído por qualquer um dos participantes [38]. Isso implica no requisito de que os classificadores que compõem o comitê devem cometer erros em diferentes partes dos dados, de modo que essas falhas tenham um impacto pequeno no resultado final retornado pelo sistema, isto é, devem apresentar diversidade [41, 160, 175].

Há alguns mecanismos importantes para estimular a diversidade em comitês de classificadores: implementar a variação nos parâmetros de funcionamento da técnicas utilizadas, a alternância na base de treinamento utilizada pelos algoritmos e a pluralidade nas técnicas de AM para a criação dos modelos de classificação [28, 37]. Em [45], o autor lista três principais justificativas para a utilização de um sistema baseado em *ensemble*, em detrimento do uso de soluções individuais de classificação:

- Estatística: considerando a existência de diferentes classificadores com bons resultados no domínio de treinamento, um único modelo escolhido pode não ter uma boa característica de generalização, ou seja, pode não obter bons resultados nos casos de teste. Ao combinar os valores de cada classificador, mitiga-se o risco de escolher um modelo inadequado;
- Computacional: muitas técnicas realizam uma busca local para a resolução do problema. Isso pode causar a estagnação dos modelos em ótimos locais, que podem estar muito longe do resultado global ótimo. A utilização de um comitê de classificadores pode prover melhores resultados, uma vez que diferentes instâncias

dos algoritmos possivelmente estarão em diferentes pontos do espaço de busca e a combinação deles pode resultar em melhores aproximações do resultado desejado;

- Representacional: dependendo da técnica utilizada, o modelo gerado pode não possuir mecanismos suficientes para representar os limites de classes de forma adequada. Os *ensembles* podem auxiliar nesse processo, uma vez que possibilita a combinação de classificadores com diferentes representações do problema, auxiliando na resolução de tarefas de alta complexidade e que não seriam tratadas adequadamente por uma solução individual.

É necessário frisar que a simples combinação de múltiplos classificadores não garante a produção de melhores predições se comparado a um dos modelos utilizados individualmente. Apesar disso, essa configuração reduz os riscos de obter os piores resultados de predição entre os participantes [175]. Adicionalmente, algumas pesquisas vem mostrando que a utilização de *ensembles* é mais eficiente que o emprego de um classificador individual em grande parte dos casos [95, 101, 45, 41].

Os *ensembles* podem ser construídos de diversas formas e podem utilizar o mesmo tipo ou tipos diferentes de técnicas de classificação. Além disso, podem variar quanto ao treinamento dos classificadores, fazendo uso da mesma base de treino para todos os modelos ou diferentes conjuntos de dados para cada um deles. Dentre as principais técnicas de comitês de classificadores disponíveis na literatura, pode-se destacar o *Bagging*, o *Boosting* e o *Stacking* [38].

No *Bagging* [26] (*Boostrap Aggregating*), a base de treinamento é dividida em N subconjuntos aleatórios, com as entradas escolhidas sendo devolvidas ao conjunto original após serem selecionadas, ou seja, uma seleção com reposição. Essa característica auxilia na diversidade dos modelos obtidos [30, 38].

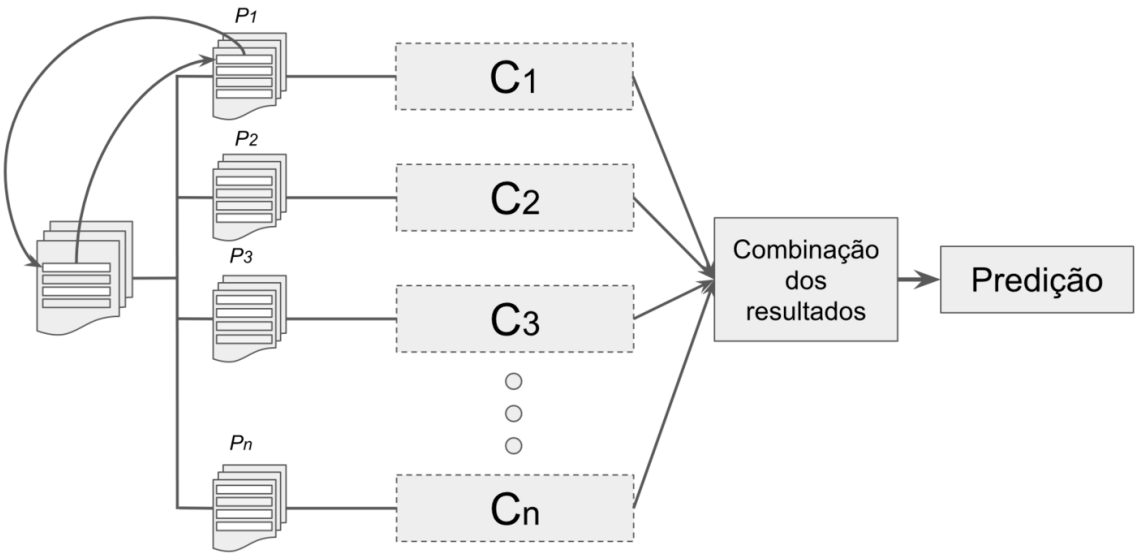

Figura 2.15: *Comitê de classificadores utilizando bagging*

A quantidade de classificadores gerados nessa técnica é igual a N e, normalmente, um esquema de votação simples é utilizado para combinar os diferentes modelos [38, 95]. Além disso, via de regra, o mesmo algoritmo é aplicado para todos os N subconjuntos. A Figura 2.15 ilustra a divisão da base original de treinamento em N partes $(P_1, P_2, ..., P_n)$, cada uma delas servindo como entrada para a criação de N classificadores $(C_1, C_2, ..., C_n)$.

Já no método *Boosting* [170], o subconjunto de treino de cada um dos classificadores de base é organizado utilizando um esquema de ponderação baseado no resultado dos primeiros modelos gerados, ou seja, os diferentes classificadores são treinados sequencialmente, sendo que C_n depende dos resultados gerados por C_{n-1}.

Essa estratégia faz com que somente o primeiro classificador da série seja criado utilizando amostragens completamente aleatórias do conjunto de treino original. Além do mais, faz com que as instâncias classificadas incorretamente tenham mais chances de serem selecionadas para o próximo classificador da série, auxiliando na criação de modelos mais abrangentes [102].

O esquema de funcionamento da técnica *boosting* pode ser visto na Figura 2.16, em que as saídas de cada um dos classificadores são utilizadas para a ponderação das mensagens originais, de forma a influenciar a escolha das entradas para o próximo classificador.

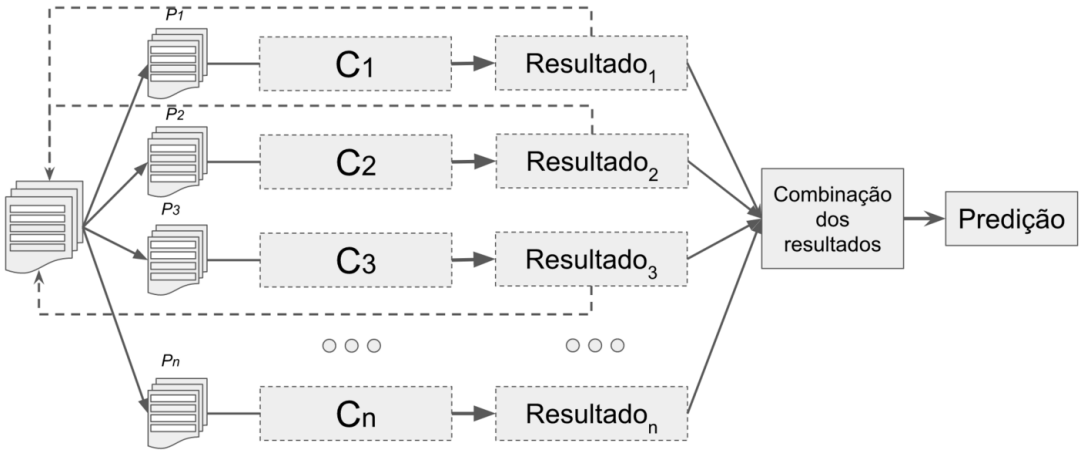

Figura 2.16: *Comitê de classificadores utilizando boosting*

A técnica *Stacking* [190], também conhecida como *Stacked Generalization*, inclui um atributo de confiabilidade para cada um dos classificadores utilizados e leva essa característica em consideração na combinação das saídas de cada modelo. Essa estratégia gera uma série de classificadores de base heterogêneos no nível 0 e os resultados desses diferentes modelos são combinados por um meta classificador, também chamado de classificador de nível 1, responsável por encontrar o melhor arranjo dos valores resultantes do nível anterior e do qual se espera um desempenho superior [65, 121].

A Figura 2.17 ilustra o funcionamento de uma estratégia de *stacking*, que utiliza uma única base de treinamento para criar N classificadores heterogêneos $(C_1, C_2, ..., C_n)$ gerados por diferentes algoritmos de AM no nível 0, com as saídas dos classificadores utilizadas como entrada para um meta classificador, responsável pela combinação dos valores para executar a predição [65, 38].

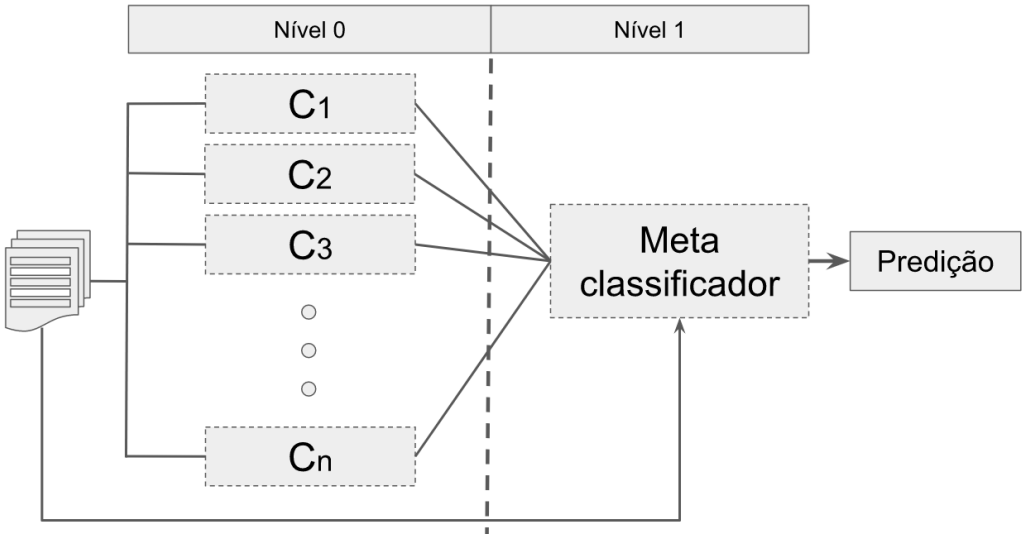

Figura 2.17: *Comitê de classificadores utilizando stacking*

As formas mais comuns de combinação das saídas dos classificadores organizados em comitê podem ser divididas em soluções algébricas e baseadas em votação [91]. As primeiras trabalham com a manipulação da probabilidade de cada uma das classes disponíveis para o problema, com o objetivo de incrementar o valor preditivo dos modelos. Como exemplo de métodos algébricos, pode-se citar a média das probabilidades (e sua versão ponderada), a soma, o produto, o máximo, o mínimo, a mediana, etc [194].

As estratégias baseadas em votação levam em consideração os rótulos de classe principal, ou seja, as predições resultantes de cada modelo. O principal método desse conjunto é a função de votação majoritária (e sua versão ponderada), que considera como resultado a classe predita pela maioria dos classificadores pertencentes ao *ensemble*.

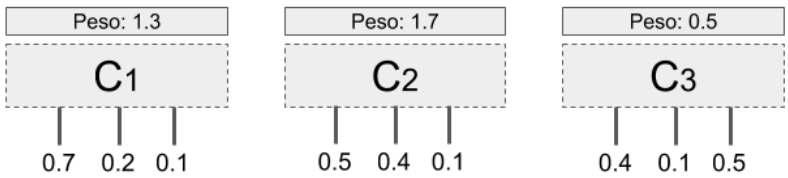

Figura 2.18: *Exemplo das principais funções de combinação*

Tabela 2.5: *Exemplo de aplicação das funções sobre classificadores*

	Probabilidade		
	Positivo	**Negativo**	**Neutro**
max(c1, c2, c3)	0.7	0.4	0.5
min(c1, c2, c3)	0.4	0.4	0.1
sum(c1, c2, c3)	1.6	0.7	0.7
avg(c1, c2, c3)	0.533	0.233	0.233
avg_w(c1, c2, c3)	0.453	0.282	0.157
majority(c1, c2, c3)		Positivo	
majority_w(c1, c2, c3)		Positivo	

Um exemplo de funcionamento dos principais métodos de combinação pode ser visto na Tabela 2.5. Nela é possível visualizar 3 classificadores $c1, c2, c3 \in C$ – demonstrados na Figura 2.18 – e suas respectivas probabilidades para as classes positiva, negativa e neutra. Além disso, cada um deles possui um peso, representando sua importância no comitê.

CAPÍTULO 3

Trabalhos Relacionados

Este Capítulo tem como propósito apresentar alguns trabalhos relevantes da área de Análise de Sentimentos, com o foco em pesquisas que utilizam o *Twitter* como fonte de dados (AST). Serão discutidos principalmente os trabalhos que aplicam abordagens híbridas, com a combinação de uma série de dicionários léxicos. Além disso, serão alvo da análise as pesquisas que combinam diversas técnicas de AM em forma de *ensembles*. Ao final do Capítulo, as principais características dos trabalhos analisados serão sumarizadas em tabelas para melhor entendimento.

Como será demonstrado nos próximos parágrafos, a quantidade de trabalhos que adotam algoritmos evolucionários como técnica para Análise de Sentimentos – em especial a Programação Genética – é muito pequena, o que demonstra que muito ainda pode ser explorado nesse sentido. O número de pesquisas na área de AS vem crescendo a cada ano, motivado, principalmente, pela importância da área no contexto atual da análise de dados. O aumento da produção de conteúdo na Internet e o fácil acesso a essas bases também permitiu que a área se expandisse consideravelmente.

A AS é uma linha multidisciplinar, que reúne áreas como Mineração de Dados, Processamento de Linguagem Natural, Recuperação de Informações, Inteligência Artificial, entre outras [20]. A maior parte das pesquisas usa abordagens baseadas em Aprendizado de Máquina Supervisionada ou baseadas em léxicos. Entretanto, alguns trabalhos apresentam resultados promissores com a utilização de abordagens híbridas, com os dicionários léxicos possuindo papel central no processo de AS [120].

Apesar do foco estar direcionado a trabalhos de AST, é importante apresentar algumas pesquisas clássicas da área de AS, que fornecem bases e definições relevantes. Um desses trabalhos formaliza matematicamente a opinião como uma quíntupla: entidade, aspecto da entidade, sentimento, autor e tempo [106]. Essa definição é utilizada em grande parte das pesquisas na área, caracterizando-se, portanto, como fundamental nos trabalhos sobre o assunto.

Outro trabalho que merece destaque pode ser lido em [182], que foca em aspectos sintáticos e semânticos do texto para a previsão de polaridades de palavras desconhecidas. Tal abordagem aplica a técnica de *Pointwise Mutual Infomation* (PMI) –

amplamente utilizada em outros trabalhos na literatura – com o objetivo de calcular a co-ocorrência de palavras e, com isso, comparar a orientação semântica de novos termos com outros previamente conhecidos, chamados de palavras-semente (*seed-words*, em inglês). Para esse trabalho, o autor faz uso das palavras *excellent* (excelente) e *poor* (pobre) representando as referências positiva e negativa, respectivamente. O teste da abordagem foi feito utilizando uma base de dados composta de 410 avaliações de produtos, obtendo uma acurácia de 74%.

No contexto de AS em redes sociais, [148] propõe uma abordagem híbrida para classificação de sentimentos de mensagens provenientes do MySpace[1]. Utiliza *Support Vector Machines* (SVM) como técnica de AM, usando como *features* o TF-IDF (*Term Frequency – Inverse Document Frequency*) [140] das mensagens, bem como as polaridades atribuídas com o auxílio do dicionário *General Inquirer*[2].

Como exemplo de trabalhos que fazem uso de outras fontes de dados e técnicas de classificação podemos citar [140], que analisa comentários sobre filmes retirados do site IMDb[3] usando SVM, *Naïve Bayes* e Regressão Logística, [129] que busca classificar comentários de *softwares*[4] e filmes fazendo uso de um dicionário manualmente anotado e SVM como técnica de AM.

A pesquisa publicada em [171] analisa notícias do mercado financeiro em busca de uma relação entre o sentimento geral sobre uma determinada empresa e o preço futuro de suas ações no mercado. Em [36], o autor faz uso da técnica de *Naïve Bayes* em conjunto com um dicionário manualmente anotado para analisar o sentimento geral de mensagens em uma sala de bate-papo de um curso de graduação a distância. O leitor interessado pode encontrar outros trabalhos da área de AS em diversas pesquisas disponíveis sobre o assunto, como [120, 80, 54, 156, 4].

3.1 Análise de Sentimentos em *Tweets*

Como discutido anteriormente, muitas pesquisas usam o *Twitter* como principal fonte de dados para a AS. Há vários motivos para o crescente número de trabalhos nessa área, dentre os quais pode-se citar a facilidade de recuperar *tweets* por tópicos (até mesmo em tempo real), a grande quantidade de usuários e, consequentemente, mensagens publicadas, a possibilidade de analisar sentimentos sobre diversos assuntos, entre outros.

No *Twitter*, é comum que as mensagens sejam escritas utilizando linguagem informal, abreviações, gírias e afins. Isso aumenta o desafio da AST, principalmente em

[1] `https://myspace.com/`
[2] `http://www.wjh.harvard.edu/~inquirer/homecat.htm`
[3] `https://www.imdb.com/`
[4] `https://www.cnet.com/topics/software/products/`

técnicas que usam dicionários léxicos, uma vez que as palavras não são armazenadas em todas as suas variações. Além disso, outras características dos *tweets* devem ser observadas, como a limitação do tamanho do texto em 280 caracteres, a grande quantidade de dados gerados, estilo informal de escrita, variedade de tópicos de discussão, entre outros [175]. Mais detalhes sobre a AST e os desafios em classificar esse tipo de mensagem são apresentados na Seção 2.1.4.

A maioria dos trabalhos de AST faz uso de um único classificador, sendo os mais utilizados o SVM, *Naïve Bayes*, Regressão Logística [175]. Como se pode perceber, não há nenhum algoritmo pertencente a classe das estratégias evolucionárias entre os mais utilizados para a AST. Muitos dos trabalhos analisados utilizam uma estratégia híbrida, com a combinação de diversos dicionários léxicos. Essas abordagens vem obtendo bons resultados, frequentemente superiores às pesquisas que utilizam AM ou métodos léxicos de forma isolada [120]. Por se tratar da estratégia adotada nesta dissertação, essa Seção terá como foco esse tipo de solução.

Um dos pioneiros na AST é o trabalho publicado em [61], que buscou classificar os *tweets* em duas classes: positiva e negativa. Para isso, o autor fez uso de técnicas clássicas de AM, como SVM, Regressão Logística e *Naïve Bayes* (esse último obtendo o melhor resultado), usando como *features* os unigramas das mensagens, bem como a frequência de palavras na representação de *bag-of-words*. O pré-processamento das entradas incluiu a supressão de letras repetidas e a exclusão dos *emoticons*. Uma evolução do experimento incluiu a classe neutra nos *tweets* e fez com que a acurácia geral do classificador diminuísse.

Uma abordagem para AST utilizando um classificador linear treinado com a técnica de *Stochastic Gradient Descent* (SGD) pode ser vista em [188]. Como atributos, utiliza o *stem*[5] da frase e a soma acumulada de polaridades positivas e negativas, com o apoio do dicionário Sentiwordnet[6]. Além disso, trabalha com 3 variantes de cada entrada: a palavra original, uma versão normalizada com todas as letras minúsculas e todos os números convertidos para 0 e uma versão com letras repetidas suprimidas. Obteve como resultado um *F1-score* de 65.54% na classificação de *tweets* presentes no *benchmark* SemEval 2013[7].

O SGD também foi utilizado para a AST em [70]. Nesse trabalho, o autor faz uso dos dicionários LIU[8], MPQA[9] (*Multi-Perspective Question Answering*) e NRC[10] para a aquisição dos valores das polaridades de cada palavra. Outras *features* importantes

[5]Processo de redução de uma palavra flexionada à sua raiz
[6]http://sentiwordnet.isti.cnr.it/
[7]https://www.cs.york.ac.uk/semeval-2013/
[8]https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#lexicon
[9]https://mpqa.cs.pitt.edu/lexicons/subj_lexicon/
[10]https://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm

utilizadas na pesquisa foram a normalização das mensagens (remoção de caracteres em sequência, separação de palavras, remoção de URLs), n-gramas de palavras, *stem*, pontuação, e características inerentes dos *tweets* como *hashtags* e menções a usuários[11]. Nos testes, a abordagem obteve um *F1-score* de 69.10% na classificação de *tweets* em positivo, negativo ou neutro.

A pesquisa publicada em [123] aborda o problema de classificação de sentimentos em *tweets* com uma estratégia híbrida fazendo uso da técnica de Regressão Logística e utilizando 7 dicionários. Como principais atributos usa a separação em n-gramas das palavras e dos caracteres, *PoS Tag* e um desambiguador de termos, além de um módulo de correção de texto para facilitar a busca nos dicionários utilizados. Utilizou como métrica o *F1-score* das classes positiva e negativa, e obteve um resultado de 70.9% em uma das bases utilizadas como teste. Em testes com mensagens de contexto geral, o classificador não obteve resultados satisfatórios.

A técnica de Regressão logística também é utilizada no trabalho publicado em [73], em conjunto com 6 dicionários: LIU, MPQA, NRC, Sentiwordnet, Sentiment140[12] e um dicionário próprio. Para o processo de treinamento do modelo, usou como *features* n-gramas da mensagem, palavras de negação e inversão de polaridades, pontuação, além do PMI das palavras [183]. Obteve como melhor resultado um *F1-score* de 64.27% em uma das bases de teste utilizada.

Ainda no contexto de utilização de Regressão Logística para AST, o trabalho publicado em [114] faz uso da técnica juntamente com um dicionário criado de forma automatizada a partir de documentos coletados da *Web* processados com o PMI [183]. Utilizando o *benchmark* SemEval 2014[13], obteve a 9ª colocação entre 50 trabalhos, classificando os *tweets* em positivo, negativo ou neutro. A técnica também é utilizada em [90] juntamente com SVM para a AST (com o último obtendo uma performance superior). Nessa pesquisa, os autores fizeram uso de diversas características sentimentais, semânticas e sintáticas das mensagens, além dos dicionários de LIU, MPQA, NRC e Sentiment140 para a aquisição das polaridades de cada uma das palavras.

Uma análise híbrida para a AST pode ser vista no trabalho publicado em [139]. Inicialmente, as mensagens são filtradas de acordo com um dicionário de *emoticons* positivos ou negativos em suas respectivas classes. Após a classificação inicial, o trabalho faz uma comparação de desempenho entre abordagens de AM: SVM, *Naïve Bayes* e *Conditional Random Field* (CRF). Os autores utilizam uma série de atributos como n-gramas, *PoS tag*, remoção de *stopwords*[14], URL e menções a outros usuários do *Twitter*.

[11] Exemplos de menções: @airtonbjunior, @twitter

[12] `http://saifmohammad.com/Lexicons/Sentiment140-Lexicon-v0.1.zip`

[13] `http://alt.qcri.org/semeval2014/`

[14] Palavras que podem ser consideradas irrelevantes para a AS

Os melhores resultados foram obtidos com o uso de *Naïve Bayes* usando *PoS tag* e n-gramas como características.

Em [196], o autor propõe uma solução híbrida com o uso de SVM e a combinação de 6 dicionários léxicos para a classificação de *tweets* em 3 classes: positivo, negativo e neutro. Utiliza como *features* palavras de negação e intensificação, n-gramas de caracteres e de palavras, *PoS Tag* e características particulares de *tweets*, como *hashtags* e palavras alongadadas[15]. A pesquisa classifica as mensagens em dois níveis de granularidade: termos e mensagens.

Outra abordagem híbrida para a AST é apresentada em [17], usando os léxicos SentiStrength[16] e LIU, além de um dicionário de palavras de negação construído manualmente. Para a criação do modelo, faz uso de SVM, utilizando como atributos palavras de negação e intensificação, n-gramas de palavras (unigrama, bigrama e trigrama), presença de *emoticons* e *hashtags*, entre outros. Obteve um *F1-score* de 63.9% na classificação, em 3 classes, de *tweets* presentes no *benchmark* SemEval 2014.

Em [87], uma abordagem de AST em 2 estágios – utilizando SVM em cada um deles – foi escolhida. No primeiro estágio, o modelo classifica a mensagem como objetiva ou subjetiva. A segunda fase tem como meta classificar as mensagens subjetivas como positivas ou negativas. O classificador faz uso de diversas *features* morfológicas, *PoS Tag* e léxicas, utilizando 11 dicionários para apoiar no processo. Obteve um *F1-score* de 64.31% na avaliação de mensagens nas classes positiva, negativa ou neutra.

O SVM também é utilizado em [18], trabalho em que o autor faz uso do dicionário MPQA e características inerentes dos *tweets* como *retweets*, *hashtags*, *emoticons*, URLs, entre outros. Obteve uma acurácia de 81,4% na classificação da polaridade de mensagens em 3 classes: positiva, negativa ou neutra.

Em [83] o autor utiliza a AST para monitorar e classificar 21.040 *tweets* sobre a gripe H1N1 na Índia, coletados entre fevereiro e março de 2015. Usa como atributos as mensagens com remoção de *stopwords*, *stem*, normalização do texto e um conjunto de palavras-chave sobre a gripe previamente levantado. As classificações foram feitas utilizando SVM, *Naïve Bayes* e *Random Forest*. Os melhores resultados foram obtidos utilizando as duas primeiras técnicas – *F1-score* de 77% – enquanto o algoritmo de *Random Forest* obteve resultados inferiores para o mesmo conjunto de dados.

Como se pode perceber, a escolha dos atributos a serem utilizados pelos métodos de classificação são essenciais para a obtenção de bons resultados. Essas características apoiam na qualidade do poder de predição desses classificadores e são, portanto, pontos importantes no projeto de um sistema de AST [175]. Alguns trabalhos, inclusive,

[15]Exemplo de palavras alongadas: Olaaaaaa mundoooo!

[16]`http://sentistrength.wlv.ac.uk/`

preocupam-se em discutir as principais *features* e o impacto delas sobre técnicas de classificação de sentimentos [2, 5, 92, 72]. Os atributos utilizados nos trabalhos relacionados são compilados e apresentados na Tabela 3.3.

A quantidade de trabalhos de AST utilizando a língua portuguesa ainda é muito pequena. Isso se deve, principalmente, ao fato da maior parte dos recursos – mensagens anotadas, dicionários, bibliotecas, etc. – estarem disponíveis em inglês. Apesar disso, é importante salientar que o Brasil possui 8.49 milhões de perfis cadastrados no *Twitter*, sendo o 6º país com mais usuários na rede[17].

Considerando o exposto no parágrafo anterior, é possível identificar alguns trabalhos recentes que fazem a AST utilizando mensagens em português. Em [53] o autor analisa *tweets* referentes à Copa do Mundo de 2014 para identificar o sentimento geral dos usuários do *site* em relação ao evento. No trabalho de [14], é feita a AST em português no contexto de saúde, em especial mensagens sobre câncer e diabetes. [39] apresenta uma abordagem utilizando um comitê de classificadores para a avaliação de *tweets* escritos em português proveninentes de uma base pré-anotada[18]. Em [10] o autor analisa a ocorrência de *bullying* contra professores no *Twitter* em português. A maior parte desses trabalhos utiliza dicionários próprios, o que comprova a limitação de recursos disponíveis para o português.

3.2 Análise de Sentimentos utilizando Computação Evolucionária

Apesar do uso de métodos tradicionais de AM serem mais comuns no campo de problemas de PLN, ainda há espaço para melhorias nos processos utilizados e nos resultados obtidos pelos trabalhos [66]. Mesmo com os resultados consolidados em outras áreas de pesquisa, a quantidade de trabalhos que utilizam métodos evolucionários em AS é pequena. Nesta Seção, apresentam-se as pesquisas mais recentes na área de AST que fazem uso de técnicas evolucionárias.

O trabalho publicado em [12] utiliza Algoritmos Genéticos (AG) para identificar a melhor combinação de 17 dicionários léxicos para um classificador de sentimentos. Utilizou como representação do cromossomo do problema um vetor de 17 posições, cada uma delas representando um dos dicionários. Qualquer um dos índices do vetor pode receber zero (correspondendo a ausência do léxico) e um (presença do léxico). Utilizando uma base de *tweets* e a técnica de Regressão Logística, obteve como melhor resultado um

[17] https://www.statista.com/statistics/242606/number-of-active-twitter-\
users-in-selected-countries/

[18] https://sites.google.com/site/miningbrgroup/home/resources

F1-score de 71.11% combinando 8 dos 17 dicionários disponíveis. O autor confirmou, portanto, que para o contexto aplicado, a simples adição de dicionários não foi suficiente para o aumento na qualidade de predição do classificador, sendo a melhor combinação representada com a ausência de 9 dos 17 léxicos.

Em [89], o autor utiliza a técnica de Algoritmos Genéticos para otimizar as polaridades de palavras de um dicionário léxico, com os valores ajustados para variarem no intervalo $[-10, 10] \in \mathbf{Z}$. O objetivo principal do trabalho é encontrar uma combinação de palavras e seus respectivos valores que minimizem o erro do classificador e, para isso, utiliza a acurácia como a função de aptidão. Para classificar as mensagens, a soma simples da polaridade de cada termo é realizado. O autor considera duas classes possíveis: positiva, caso o resultado do classificador seja maior que zero, e negativa, caso o resultado seja menor ou igual a zero. Obteve como melhor resultado um *F1-score* de 84.78% em uma base de dados contendo opiniões sobre debate político[19] [44].

A pesquisa publicada em [1] usa um método híbrido contendo Algoritmos Genéticos para a seleção de atributos em discussões de fóruns na *Web*. Em [19], o método de Otimização por Enxame de Partículas (*Particle Swarm Optimization* – PSO) é utilizado em conjunto com SVM para a análise de opiniões sobre avaliações de filmes postados no *Twitter*. Ainda no contexto de avaliação de filmes, [64] apresenta uma solução híbrida, fazendo uso de *Naïve Bayes* e Algoritmos Genéticos, usando como base de dados 2.000 mensagens disponíveis para *download*[20].

Especificamente no contexto de utilização de Programação Genética para a AS, cita-se o trabalho de [66] que faz uso de uma variação da PG criada por [128] chamada Programação Genética Semântica (PGS), que considera a representação fenotípica dos indivíduos na utilização dos operadores genéticos, buscando otimizar a geração de descendentes. Essa pesquisa propõe uma nova técnica chamada *Root Genetic Programming*, em que as operações genéticas são aplicadas somente na raiz da árvore. Usando como *benchmark* os dados fornecidos pela SEPLN[21] (*Sociedad Española para el Procesamiento del Lenguaje Natural* – Sociedade Espanhola para o Processamento de Linguagem Natural), tem como objetivo a classificação de 7.218 mensagens em 6 classes – muito positiva, positiva, neutra, negativa, muito negativa e *None* (ausência).

Os autores adotaram uma estratégia um-contra-um (*one-against-one*) [75, 110], ou seja, foram criados classificadores binários para a distinção entre cada uma das 6 classes (resultando em 15 classificadores totais). A representação das mensagens de entrada foi feita por meio de um vetor numérico criado com a utilização da técnica de TF-IDF (detalhada no Capítulo 2).

[19] SOMD - *Strict Obama McCain Debate dataset*

[20] `http://www.cs.cornell.edu/people/pabo/movie-review-data/`

[21] `www.sepln.org/workshops/tass/2015/tass2015.php`

Aplicando uma estratégia gulosa de *crossover*, onde os filhos resultantes da operação só são mantidos se apresentarem *fitness* melhores que os dos pais, o trabalho mostrou-se competitivo com outros classificadores utilizados na AST – SVM, *Naïve Bayes*, *Adaboost* e *Extreme Random Tree* – apresentando resultados inferiores somente com relação à técnica de SVM.

Em comparação com a proposta desta dissertação (detalhada no Capítulo 4), destaca-se, como principais diferenciais, a utilização da Programação Genética definida em [93], sem modificações nos operadores genéticos principais nem restrições quanto ao *crossover*. Além disso, a representação da mensagem é feita em seu formato textual original, ao invés da sua conversão em um vetor numérico, conforme utilizado em [66].

Essa última característica reflete diretamente no conjunto de funções utilizado para a manipulação dos indivíduos e, em nossas pesquisas, não foram encontrados trabalhos que utilizem esse tipo de representação para a AST com a utilização de PG. É possível destacar, ainda, a diferença na estratégia de classificação: enquanto o presente trabalho cria um único classificador ternário (positivo, negativo e neutro), a pesquisa publicada em [66] faz uso de 15 classificadores binários para a avaliação de cada uma das mensagens nas 6 classes disponíveis. O tipo de seleção e a forma de criação de indivíduos são os mesmos em ambos os trabalhos: torneio com $k = 2$ e *Ramped half-and-half*, respectivamente.

A utilização da PG para a definição dos intervalos de valores de cada uma das classes e a ponderação dos dicionários léxicos (explanadas em detalhes no Capítulo 4, nas Seções 4.5 e 4.7, respectivamente) também são diferenças relevantes em comparação com a pesquisa discutida nos parágrafos anteriores. Essa última característica auxilia na busca pela validação da Hipótese de Pesquisa 2, que afirma que o modelo de classificação de sentimentos híbrido, criado com a utilização da PG, é capaz de determinar a relevância dos dicionários empregados no sistema.

Como se pode perceber, a Programação Genética, apesar de obter resultados significativos em diversas áreas de pesquisa, raramente é utilizada no contexto de Processamento de Linguagem Natural (PLN), em especial na AS e AST. Vale destacar que um dos principais diferenciais da abordagem utilizando a PG é a possibilidade de customização ou generalização – dependendo da base de treinamento e funções utilizadas – e também um entendimento de como o modelo atribui classe para as mensagens. As soluções geradas podem ser lidas e interpretadas pelos usuários, uma vez que a representação do modelo produzido pela PG é transparente [96].

Além do exposto no parágrafo anterior, devido a forma com que realiza a combinação de funções e terminais em suas soluções, a Programação Genética pode ser vista como uma ferramenta de seleção automatizada de *features* para processos de classificação [66], uma vez que a estrutura geral da solução também é parte integrante

do processo de busca [136], e características que não agreguem valor ao modelo serão descartadas pelo processo de evolução da PG por implicarem em uma função de aptidão baixa. Outros benefícios da utilização da Programação Genética são descritos no Capítulo 1 e nos trabalhos de [93, 145].

Com o objetivo de sumarizar o exposto neste Capítulo, informações relevantes dos principais trabalhos relacionados foram organizadas em tabelas, facilitando a análise. As técnicas utilizadas por pesquisas da área de AST são apresentadas na Tabela 3.1. Os trabalhos analisados mostram que a maior parte das pesquisas utilizam a técnica de SVM e suas variações e parametrizações. O algoritmo de *Naïve Bayes* também vem sendo amplamente utilizado, obtendo bons resultados em suas predições.

Na Tabela 3.2 são apresentados os dicionários léxicos utilizados pelas principais pesquisas e é possível observar a tendência das soluções em empregar mais de um dicionário em suas abordagens. Da mesma forma, os principais atributos usados no processo de AST por meio de AM são apresentados na Tabela 3.3. Essas informações foram utilizadas como base para a escolha das *features* deste trabalho.

Tabela 3.1: *Técnicas de Aprendizado de Máquina utilizadas em trabalhos relacionados*

Técnica	**Trabalhos**
Support Vector Machines (SVM)	[148, 140, 129, 19, 139, 87, 18, 83]
Naïve Bayes	[140, 64, 139, 83]
Regressão Logística	[140, 90, 123, 73]
Conditional Random Field (CRF)	[139]
Stochastic Gradient Descent	[188, 70]
Random Forest	[83]
Algoritmos Genéticos	[12, 89, 1, 64]
Programação Genética	[66]

Tabela 3.2: *Dicionários Léxicos utilizados em trabalhos relacionados. Na Tabela, NRC_e remete ao dicionário de emoticons NRC, NRC_h ao dicionário NRC de hashtags, SWordnet ao léxico Sentiwordnet e S140 ao dicionário Sentiment140*

	Dicionário							
	LIU	MPQA	NRC_e	NRC_h	SWordnet	AFINN	S140	Léxico Próprio
[196]	✓	✓	✓	✓	✓	✓		✓
[17]	✓				✓			
[148]								✓
[114]			✓		✓		✓	✓
[123]	✓	✓	✓	✓	✓	✓	✓	
[87]	✓	✓	✓	✓	✓	✓	✓	✓
[100]	✓	✓	✓	✓		✓	✓	✓
[164]	✓	✓				✓		✓
[55]	✓		✓	✓			✓	
[82]	✓	✓	✓	✓			✓	
[70]	✓	✓		✓				
[181]	✓	✓	✓	✓			✓	✓
[51]	✓	✓		✓		✓	✓	✓
[17]	✓							
[6]	✓	✓		✓				✓
[184]	✓	✓	✓					✓
[12]	✓		✓		✓		✓	
[90]	✓	✓	✓				✓	✓
[73]	✓	✓		✓	✓		✓	✓
[69]					✓			

Tabela 3.3: *Atributos utilizados em trabalhos relacionados. Na Tabela, PoS faz referência à Part-of-speech, Neg à negação, Intens à intensificação, Pont à pontuação, Rep à repetição*

	Feature							
	PoS	Neg	Intens	N-gram	Pont	Rep	Maiúsculas	Léxico
[196]	✓	✓	✓	✓	✓		✓	✓
[17]	✓	✓		✓	✓	✓	✓	✓
[148]	✓				✓			✓
[114]	✓	✓		✓	✓	✓	✓	✓
[123]	✓	✓		✓				✓
[87]	✓		✓	✓	✓	✓	✓	✓
[100]	✓	✓		✓	✓	✓	✓	✓
[164]	✓	✓		✓	✓	✓	✓	✓
[55]	✓	✓	✓	✓	✓	✓		✓
[82]	✓	✓		✓				✓
[70]	✓	✓	✓	✓	✓	✓	✓	✓
[181]		✓		✓	✓	✓	✓	✓
[51]		✓		✓				✓
[17]	✓	✓				✓	✓	✓
[6]		✓						✓
[184]	✓	✓		✓	✓	✓	✓	✓
[12]		✓		✓				✓
[90]	✓	✓		✓	✓	✓	✓	✓
[73]	✓	✓				✓	✓	✓
[69]				✓	✓	✓		✓

3.3 Análise de Sentimentos utilizando Comitê de Classificadores

A utilização de comitês de classificadores vem sendo pouco explorada na área de Análise de Sentimentos, em especial na AST. Apesar disso, é possível identificar alguns esforços empreendidos nos últimos anos para mudar esse cenário. Entre esses trabalhos, cita-se a pesquisa publicada em [37], que cria um *ensemble* de classificadores para a AST utilizando diferentes técnicas, a saber: *Random Forest*, *Support Vector Machines* (SVM), *Naive Bayes* e Regressão Logística. As estratégias de combinação utilizadas foram a média das probabilidades de cada classificador e o voto majoritário e, com isso, uma acurácia de 76% foi obtida em uma das bases de teste.

Em [103] o autor apresenta uma proposta de comitê utilizando somente a técnica de Regressão Logística, variando o tamanho dos *ensembles*, as *features* utilizadas por cada modelo e o conjunto de mensagens de treinamento. Os resultados mostraram que a utilização dos classificadores em conjunto melhorou a acurácia geral de classificação de *tweets*. Além disso, o emprego de diferentes bases de treinamento, ao invés de subconjuntos aleatórios de uma mesma base para cada um dos modelos, mostrou ser eficaz para o aumento da capacidade de predição do sistema. Em seus melhores resultados, o trabalho obteve uma acurácia de 80% nas mensagens avaliadas.

Na mesma linha, [34] propõe um esquema de *ensemble* para a AST utilizando apenas uma técnica de AM, o *Naïve Bayes*. O trabalho faz a seleção de uma série de atributos – n-gramas, léxicos (MPQA), *Pos Tag* e *tokens* especiais – e treina cada classificador utilizando apenas uma das *features*. Para a combinação dos modelos, adota uma estratégia de votação ponderada. Obteve um *F1-score* de 67% na avaliação de *tweets* em positivo e negativo.

Em [74], o autor apresenta uma solução de *ensemble* utilizando as técnicas de Redes Neurais, *Naïve Bayes*, Regressão Logística e SVM para análise de *tweets*. A proposta busca garantir a diversidade dos modelos gerados combinando diferentes conjuntos de *features* – dicionários léxicos (Sentiwordnet), *PoS tag*, n-gramas de palavras – além de diferentes conjuntos de treinamento e parâmetros dos algoritmos. Nos melhores resultados, a técnica obteve uma acurácia de 71%, obtendo o 3º lugar entre 8 abordagens avaliadas.

Como é possível observar, nenhuma das técnicas discutidas fez uso da Programação Genética como um dos classificadores de base. Além disso, a maioria das pesquisas utiliza uma variação no conjunto de atributos ou na base de treinamento como forma de obter diversidade.

Como será demonstrado no Capítulo 5, com o objetivo de validar a Hipótese de Pesquisa 3, que afirma que o uso da PG em conjunto com outras técnicas de AM pode

incrementar o poder de predição e o resultado geral do sistema de AS, será implementado um comitê de classificadores por meio da combinação de diferentes algoritmos de AM, garantindo a diversidade com a escolha de técnicas distintas [28]. Além disso, por se tratar de uma abordagem pouco utilizada na AS, a PG reforça manutenção da pluralidade no *ensemble*.

CAPÍTULO 4

Programação Genética na Análise de Sentimentos

Neste Capítulo são apresentadas, de forma geral, as propostas do trabalho. As etapas e informações relevantes para o entendimento da solução são detalhadas, de forma a apoiar no entendimento e acompanhamento dos próximos Capítulos, em especial o Capítulo 6 que trata dos resultados desta pesquisa.

O desafio de gerar um classificador de sentimentos pode ser descrito como um problema de otimização, com o objetivo de encontrar um modelo que represente a solução desejada. Ao abordar o problema dessa maneira, pode-se fazer uso de qualquer método de Aprendizado de Máquina capaz de gerar modelos de classificação. A Programação Genética, explanada em detalhes no Capítulo 2, é uma dessas técnicas e é a abordagem escolhida para a solução.

Apesar do uso de métodos tradicionais de Aprendizado de Máquina serem mais comuns no campo de problemas de PLN, ainda há espaço para melhorias nos processos utilizados e nos retornos obtidos pelos trabalhos [66]. Mesmo com os resultados consolidados em outras áreas de pesquisa, a quantidade de trabalhos que utilizam a Programação Genética na área de Análise de Sentimentos é muito pequena, com algumas exceções [66, 16].

Ao utilizar a PG para a resolução de problemas, deve-se iniciar com os passos preparatórios [93, 145]. Nessa etapa, características importantes do escopo do problema e da solução são elicitadas. Uma configuração inconsistente nessa fase pode prejudicar todo o processo de descoberta do resultado. Dentre as principais decisões tomadas nessa etapa, destacam-se a definição do conjunto de funções e terminais do problema, a função de aptidão, parâmetros de controle e critérios de parada.

Para tratar a questão de geração do classificador de sentimentos como um problema de otimização, considera-se M como o conjunto de modelos possíveis, sendo que cada modelo $m \in M$ é composto de elementos $e \in E$ que analisam aspectos da mensagem sob classificação. Sendo $f_m(x)$ a função que avalia x em cada modelo m, o objetivo da otimização é encontrar um modelo m' tal que sua função $f_{m'}(x) > f_m(x) \; \forall \; f_m(x)$, maxi-

mizando o resultado dos modelos.

4.1 Funções e terminais

Conforme a sequência dos passos preparatórios para a utilização da PG na resolução de problemas [93, 145], é necessário definir o conjunto F de funções f e o conjunto T de terminais t para o domínio em questão. Como se está em busca de um modelo de classificação de sentimentos em textos, é essencial a definição de funções que manipulam esse tipo de informação. Além disso, métodos matemáticos e condicionais foram incluídos para a possível utilização pelos modelos. As principais funções utilizadas para o problema são apresentadas na Tabela 4.1 e o conjunto T de possíveis terminais t é formalizado em 4-1.

$$T = \{tweet, \delta\}, -2 \leq \delta \leq 2 \tag{4-1}$$

Tabela 4.1: *Principais funções da PG utilizadas no trabalho*

ID	Função	Descrição
1	polSum(msg)	Soma das polaridades das palavras
2	polSumAVG(msg)	Média aritmética das polaridades
3	polSumAVGW(msg, $[w_1...w_n]$)	Média ponderada das polaridades
4	hashtagPolSum(msg)	Soma das polaridades das *hashtags*
5	emoticonPolSum(msg)	Soma das polaridades dos *emoticons*
6	hasHashtags(msg)	Checa se há *hashtags* na mensagem
7	hasEmoticons(msg)	Checa se há *emoticons* na mensagem
8	hasURL(msg)	Checa se há URL na mensagem
9	hasDate(msg)	Checa se há data na mensagem
10	if_then_else(*bool*, c1, c2)	Se bool é verdadeiro então c1 senão c2
11	removeStopwords(msg)	Remove *stopwords* da mensagem
12	spellCheck(msg)	Realiza a correção gramatical da mensagem
13	neutralRange(IR, SR)	Limite inferior/superior para classe neutra
14	negationWords(msg)	Considera as palavras de negação
15	boosterWords(msg)	Considera as palavras de intensificação
16	upperCaseWords(msg)	Considera as palavras em maiúsculo
17	wordCount(msg)	Quantidade de palavras
18	msgLength(msg)	Tamanho da mensagem
19	add, sub, mul, div, sen, cos, exp	Funções matemáticas

Como é possível observar, o conjunto T de terminais t é composto do *tweet* – entrada da PG – e de uma constante δ. Não há funções com aridade zero para a solução proposta. É importante frisar que o parâmetro de entrada das árvores criadas pela PG é composto da mensagem original, sem nenhuma manipulação. Essa estratégia tem por objetivo permitir que o próprio algoritmo identifique as principais alterações necessárias no texto, priorizando a composição de funções que resultem em uma maior aptidão.

É sabido, entretanto, que algumas funções de normalização de texto são amplamente utilizadas por trabalhos de AST e, por isso, foram incluídas no conjunto F, conforme detalhado a seguir. É relevante salientar essa característica de seleção automatizada de *features* da PG, que pode demonstrar quais modificações no texto original são mais expressivas para a maximização do poder de predição do modelo de AST, podendo servir como uma forma de validação de conceito.

4.1.1 Funções léxicas

As funções léxicas fazem uso dos dicionários para buscar a polaridade de cada uma das palavras da mensagem avaliada. De forma geral, a mensagem é representada no formato de *tokens* – sequência de termos – e é realizada uma busca em cada um dos léxicos de forma a identificar o valor de polaridade do termo em cada um deles.

Na função $polSum(msg)$, que retorna um número real, é feita uma soma simples das polaridades das palavras encontradas em cada um dos dicionários, conforme apresentado na Equação 4-2. O sentimento da mensagem m é calculado como a soma da polaridade de cada termo j em cada um dos dicionários i de um total de n termos e d dicionários utilizados. O Algoritmo 3 apresenta um pseudocódigo do funcionamento da função para melhor entendimento.

$$sentimento_m = \sum_{i=1}^{d} \sum_{j=1}^{n} pol_{j,i} \tag{4-2}$$

Algorithm 3 Algoritmo de soma de polaridades polSum

```
soma_polaridade ← 0
palavras ← msg {retorna todas as palavras da mensagem}
for all palavras do
    soma_polaridade += polaridade(palavra, dicionario) {polaridade da palavra}
end for
return soma_polaridade
```

A função $polSumAVG(msg)$, por sua vez, considera a média aritmética das polaridades de cada palavra, utilizando a fórmula apresentada na Equação 4-3. O método,

assim como o anterior, recebe a mensagem como parâmetro, e retorna um valor real.

$$sentimento_m = \frac{1}{d}\sum_{i=1}^{d}\sum_{j=1}^{n} pol_{j,i} \tag{4-3}$$

Na mesma linha, a função $polSumAVGW(msg, [w_1...w_n])$ calcula a média ponderada das polaridades, ou seja, cada um dos dicionários possui um peso, que representa sua importância na avaliação geral, conforme demonstrado pela Equação 4-4. O método, assim como os anteriores, recebe como parâmetro a mensagem a ser avaliada, além de *n* valores reais, com *n* igual ao número de dicionários utilizados para o problema. Os léxicos utilizados neste trabalho e seus respectivos atributos de peso são apresentados na Tabela 4.6.

$$sentimento_m = \frac{1}{\sum_{i=1}^{d} w_i} \times \sum_{i=1}^{d}\sum_{j=1}^{n} pol_{j,i} \times w_i \tag{4-4}$$

Variações das funções apresentadas anteriormente são encontradas nos métodos $emoticonPolSum(msg)$ e $hashtagPolSum(msg)$ que realizam a soma das polaridades dos *emoticons* e das *hashtags* das mensagens, respectivamente. Ambas recebem como parâmetro a mensagem a ser avaliada e retornam um valor real.

Vale comentar que, apesar de simples, a abordagem de soma das polaridades das palavras e suas variações são amplamente utilizadas na literatura, principalmente em trabalhos de AST utilizando as estratégias léxica e híbrida, e vem obtendo resultados relevantes [46, 127, 89, 42, 3, 130, 180].

4.1.2 Funções de transformação de mensagens

As funções de transformação de mensagens manipulam o texto passado como parâmetro e retornam uma *string* modificada. Importante salientar que, considerando o domínio ao qual este trabalho está inserido e os recursos utilizados – *benchmark* (Seção 5.1), dicionários (Seção 4.5) e bibliotecas (5.2) – todas as transformações sintáticas são realizadas considerando o idioma inglês.

A função $removeStopWords(msg)$ remove as *stopwords* da mensagem, retornando a entrada sem elas. Essas palavras são frequentemente ignoradas pelos trabalhos de AS, por não possuírem, muitas vezes, valor para as análises. Exemplos de *stopwords* disponibilizadas pela biblioteca NLTK[1] são *"I", "you", "ourselves", "hers", "but", "again", "there", "out", "having", "with", "they", "own", "an", "be", "some", "for", "do", "its", "of", "my"*, etc.

[1] `https://www.nltk.org/`

A título de exemplo, o processo de manipulação de uma mensagem ("*I love you my dear!*" – "Eu amo você meu querido!" em português) pela função $removeStopWords(msg)$ é apresentado no Exemplo 5:

Exemplo 5:
Mensagem original: "*I love you my dear!*"
Mensagem sem *stopwords*: "*love dear!*"

Além da função anterior, há outras disponíveis para a utilização da PG e que tem por objetivo a remoção de algum elemento da entrada. O método $removePonctuation(msg)$ elimina toda a pontuação da mensagem passada como parâmetro. Já a função $removeURL(msg)$ remove as URLs do argumento de entrada.

A função $negationWords(msg)$ tem por objetivo identificar as entradas de negação (explicadas na Seção 2.1.1) da mensagem. Essas palavras são marcadas com uma *tag* "_NEG"e são utilizadas para a inversão de polaridades de termos relacionados. Recebe como parâmetro a mensagem a ser avaliada e retorna-a com as palavras de negação identificadas e marcadas. Exemplos de entradas de negação são *"no", "don't", "doesn't", "aren't"*, etc. O Exemplo 6 apresenta uma mensagem ("*I didn't like this phone!*" – "Eu não gostei deste telefone!" em português) avaliada pela função.

Exemplo 6:
Mensagem original: "*I didn't like this phone!*"
Marcação de negação: "*I* $\boxed{\textit{didn't\_}\textbf{\textit{NEG}}}$ *like this phone!*"

De forma semelhante, o método $boosterWords(msg)$ identifica palavras de intensificação. Essas entradas servem como referência para um incremento da polaridade das frases avaliadas. Exemplos de termos de intensificação são *"very", "much", "deep"*, etc. O Exemplo 7 apresenta uma mensagem ("*I'm very frustrated with this phone*" – "Eu estou muito frustrado com este telefone" em português) processada pelo método.

Exemplo 7:
Mensagem original: " *I'm very frustrated with this phone*"
Marcação de intensificação: " *I'm* $\boxed{\textit{very\_}\textbf{\textit{BOOST}}}$ *frustrated with this phone*"

Ao utilizar a função $stemmingWords(msg)$ as palavras são flexionadas para sua forma raíz. Por exemplo, caso seja encontrada a entrada "*cats*", o processo de *stemming* flexionará para "*cat*". Esse processo é muitas vezes utilizado em abordagens que dependem de dicionários léxicos, uma vez que é muito custoso manter todas as variações de um termo.

4.1.3 Funções de verificação

Essas funções verificam algum atributo da mensagem e retornam um valor booleano identificando a presença (verdadeiro) ou ausência (falso) da caracterísitca desejada.

As funções $hasEmoticons(msg)$, $hasHashtags(msg)$, $hasURL(msg)$ e $hasDate(msg)$ verificam se a mensagem possui *emoticon*, *hashtags*, URL e datas, respectivamente.

4.1.4 Função condicional

A função $if\_then\_else(bool, c1, c2)$ recebe como parâmetro uma condição a ser verificada e dois atributos, representando a ação caso a condição seja verdadeira e outra caso seja falsa, respectivamente. Esse método tem por objetivo permitir que a PG forme componentes lógicos para a classificação das mensagens. Isso é possível fazendo uso de funções de verificação, por exemplo.

Em uma situação hipotética em que o modelo queira executar a soma de polaridades de *emoticons* – se a mensagem possuir essa característica – e, caso contrário, realizar a soma de polaridade das palavras, a PG poderia criar um encadeamento de funções do tipo $if\_then\_else(hasEmoticons(msg), emoticonPolSum(msg), polSumAVG(msg))$. A visualização desse arranjo em formato de árvore pode ser visto na Figura 4.1 e a sua representação em pseudocódigo é apresentado no Algoritmo 4.

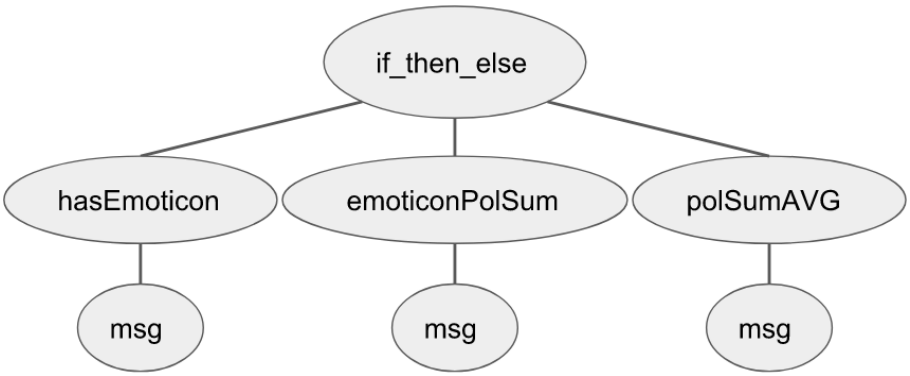

Figura 4.1: *Exemplo de uso da função condicional na Programação Genética*

Algorithm 4 Exemplo de uso da função condicional na Programação Genética

```
if hasEmoticon(msg) then
    return emoticonPolaritySum(msg)
else
    return polSumAVG(msg)
end if
```

4.2 Função objetivo

Os dois primeiros passos preparatórios da PG [93] definem as primitivas P da técnica e, com isso, os espaços que os modelos poderão explorar por meio da combinação das funções $f \in F$ e terminais $t \in T$ das possíveis soluções. Entretanto, nessa etapa ainda não é possível determinar quais desses elementos estão próximos do resultado pretendido. Para isso, deve-se definir a função objetivo, que busca representar, de forma quantitativa, a distância de um indivíduo em relação a solução desejada ou quão boa é essa solução, considerando o domínio do problema [145, 147].

É importante salientar que a função *fitness* da PG possui algumas particularidades que a diferencia das funções objetivo de outros algoritmos, inclusive da família das estratégias evolucionárias. Como os indivíduos criados pela PG são programas – ou modelos – devem ser executados para que a técnica possa avaliar a qualidade desses por meio da função de aptidão. Assim, frequentemente, todos os programas da população devem ser executados múltiplas vezes [145].

A métrica utilizada como função de aptidão e para a avaliação dos modelos da PG neste trabalho foi o *F1-score* médio das classes positivas e negativas [132]:

$$F_1^{PN} = \frac{F_1^P + F_1^N}{2}. \tag{4-5}$$

As classes disponíveis são:

$$X = \{Positivo, Negativo, Neutro\}. \tag{4-6}$$

O *F1-score* de uma classe $x \in X$ é calculado como a média harmônica entre Precisão (π) e Revocação (*Recall*) (ρ) dessa classe. Essa métrica dá uma boa ideia da qualidade do classificador, principalmente em avaliações de bases desbalanceadas, como é o caso deste trabalho (como demonstrado no Capítulo 5, nas Tabelas 5.1 e 5.2).

$$F_1^x = \frac{2\pi^x\rho^x}{\pi^x + \rho^x}. \tag{4-7}$$

A Precisão e o *Recall* de uma classe x são são obtidos com:

$$\pi^x = \frac{PP}{PP + PU + PN}. \tag{4-8}$$

$$\rho^x = \frac{PP}{PP + UP + NP}. \tag{4-9}$$

sendo PP, PN, NP, PU e UP obtidos na matriz de confusão, como demonstrado na Tabela 4.2.

Tabela 4.2: *Matriz de confusão*

		Classe Real		
		Positivo	Neutro	Negativo
Predição	Positivo	PP	PU	PN
	Neutro	UP	UU	UN
	Negativo	NP	NU	NN

Além disso, para todos os modelos, a métrica de acurácia (β) também foi calculada, conforme a Equação 4-10.

$$\beta = \frac{PP + UU + NN}{TotalMensagens}. \tag{4-10}$$

A acurácia indica a porcentagem de mensagens classificadas corretamente. Isoladamente, pode não ser uma métrica eficaz para o classificador uma vez que, caso a classe de maior ocorrência seja conhecida, basta atribuir esse valor para todas as mensagens com o objetivo de obter bons resultados [22].

A Precisão evidencia a porcentagem de mensagens classificadas corretamente para determinada classe. Assim como a acurácia, a precisão pode não refletir o desempenho geral do classificador a contento, pois mesmo que a maioria das mensagens de determinada classe não sejam avaliadas, basta que a maior parte das avaliações da mesma sejam feitas de forma correta para que se obtenha um valor de precisão alto [22].

O *Recall* é calculado avaliando a razão entre o total de mensagens de determinada classe avaliada corretamente pelo total de mensagens dessa classe. Essa métrica pode identificar indícios de uma distorção entre o valor da precisão e o resultado geral do classificador [22].

Uma forma de identificar se as diferenças obtidas entre os resultados de dois classificadores são estatisticamente significativas é por meio da utilização do teste t pareado (detalhes sobre a medida podem ser lidos em [78]). Esse cálculo pode determinar se as diferenças são nulas ou não, de acordo com um nível de significância. A fórmula para o cálculo de t é apresentada na Equação 4-11, onde m representa a média das diferenças entre os valores, d é o desvio padrão das diferenças e n é a quantidade de amostras.

$$t = \frac{m}{\frac{d}{\sqrt{n}}} \tag{4-11}$$

Para a utilização dessa métrica, há duas hipóteses disponíveis [158]:

- Hipótese nula:

  $H_0 : \mu_1 = \mu_2$: a média das populações analisadas são iguais.

- Hipótese alternativa:

  $H_1 : \mu_1 \neq \mu_2$: a média das populações analisadas são diferentes.

Neste trabalho, busca-se determinar se $H_1 : \mu_1 \neq \mu_2$. O nível de significância, identificado pela letra α, representa a taxa tolerável de erro e frequentemente é fixada em 0.05, ou seja, 5% [78]. Com os valores de t e α é possível constatar se os resultados são estatisticamente significantes por meio da consulta do valor em uma tabela de distribuição t[2] – tabela que contém os valores críticos para cada uma das combinações de t, α e grau de liberdade g, esse último definido como $n-1$.

A taxa α definida para este trabalho foi fixada em 5%, ou seja, pode-se rejeitar a Hipótese nula H_0 em caso de $p \leq 0.05$. Dado o exposto, também é possível afirmar que a confiança adotada para os testes é de 95%. É importante salientar que, nos casos em que a diferença não é estatisticamente significativa, não pode-se afirmar que não há diferenças entre as entradas, mas tão somente que não há evidências para rejeitar a Hipótese nula H_0, ou seja, o resultado pode ser considerado como inconclusivo quanto à significância [49].

4.3 Parâmetros gerais de controle

Na sequência dos passos preparatórios para a utilização da PG [93], a definição dos parâmetros gerais da técnica é de extrema importância para a resolução do problema. Como citado no Capítulo 2, esses dados dependem fortemente do domínio em que a PG está sendo aplicada, e não há configurações padrão para essas características. Apesar disso, alguns autores definem valores de referência para alguns tipos de problemas que já foram amplamente explorados com a PG [93, 145]. Os parâmetros gerais utilizados neste trabalho são apresentados na Tabela 4.3.

Apesar de alguns autores argumentarem que a melhor configuração para a maioria dos problemas possui uma quantidade pequena de gerações (frequentemente 51) e uma população substancialmente maior [93], em testes empíricos foi possível perceber que, para o problema em questão, haviam melhorias significativas nos melhores indivíduos em gerações muito próximas do limite de 51, reforçando a hipótese de que um aumento na quantidade de gerações poderia acarretar em modelos melhores.

[2] http://www.sjsu.edu/faculty/gerstman/StatPrimer/t-table.pdf

Tabela 4.3: *Parâmetros gerais da Programação Genética utilizados no trabalho*

Parâmetro	**Valor**
Prob. cruzamento	90%
Prob. mutação	5%
População	250
Gerações	150
Fitness	*F1-score*
Criação população inicial	
Tipo	*Ramped half-and-half*
Altura árvore	Entre 1 e 3
Método de Seleção	
Tipo	Torneio
Tamanho (k)	2
Condição de parada	# de gerações
Altura máxima da árvore	15

Um exemplo de evolução de dois modelos obtidos a partir do conjunto de dados utilizado nos experimentos (apresentado em detalhes no Capítulo 5) usando a configuração supracitada (51 gerações) pode ser visto na Figura 4.2. Nesse caso, é possível perceber que o melhor valor de *fitness* para o Modelo 1 foi obtido na geração 48 de 51, e do Modelo 2 na geração 49.

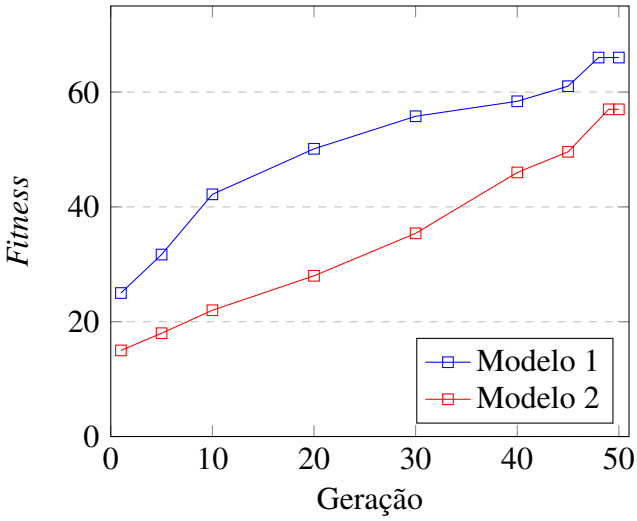

Figura 4.2: *Exemplo de evolução do fitness de dois modelos gerados*

Como se pode perceber, os melhores valores dos indivíduos foram alcançados

em gerações muito próximas do limite de 51 gerações definido inicialmente, o que reforçou a hipótese apresentada no parágrafo anterior de que um acréscimo na quantidade de gerações poderia aperfeiçoar ainda mais o modelo. O tamanho da população e a quantidade de gerações foram definidos, portanto, após testes empíricos preliminares.

Importante salientar, ainda, que é possível encontrar na literatura diversos trabalhos que utilizam uma configuração com número de gerações maior que o de indivíduos para a resolução de problemas por meio da PG [107, 186, 145].

A probabilidade de cruzamento e mutação foram definidas em 90% e 5%, respectivamente. Esses valores foram estabelecidos após testes de verificação do comportamento do algoritmo e por meio de pesquisas na literatura, apresentando-se como os valores utilizados com mais frequência pelos trabalhos [93, 145].

A escolha do Torneio como método de seleção tem por objetivo evitar que os melhores indivíduos monopolizem o processo, principalmente após algumas gerações [145]. Devido a sua forma de funcionamento (demonstrada no Capítulo 2), a técnica evita que esse problema aconteça, auxiliando na manutenção da diversidade genética da população. O parâmetro k do Torneio – quantidade de indivíduos selecionados para cada rodada – foi definido como 2 após testes empíricos e pesquisas na literatura.

A criação da população da PG é feita por meio do método *Half-and-half* pois, além de ser a técnica mais utilizada na literatura [145, 93], gera indivíduos heterogêneos, apoiando na diversidade da população inicial.

Para o problema em questão, 2 critérios de parada foram definidos: quantidade máxima de gerações obtida e função de aptidão com 100% de acerto. Qualquer uma dessas situações faz com que o algoritmo pare com o processo de evolução e retorne o melhor indivíduo como resultado.

4.4 Restrições

Métodos de penalização são as formas mais comuns de restrições em estratégias evolucionárias [31]. Esses processos têm por objetivo identificar e evitar comportamentos não desejados, transferindo para a avaliação dos indivíduos um possível conhecimento prévio sobre o domínio do problema. Neste trabalho, foram definidos processos de penalização para a repetição de funções específicas na árvore, bem como para algumas combinações indesejadas de parâmetros nos métodos.

A primeira restrição tem relação com os parâmetros da função *neutralRange(IR, SR)* (ID 10 da Tabela 4.1). Caso o valor do nível inferior (*IR*) seja maior que o nível superior (*SR*), aplica-se uma penalização no *fitness* do indivíduo e a atribuição do valor zero para ambos os parâmetros.

A repetição de algumas funções nos indivíduos não trazem melhorias nos resultados e, algumas vezes, causam inconsistências na definição de outros atributos. Além disso, funções como a 1, 2 e 3 da Tabela 4.1 são massivas e consomem muito processamento, pois iteram sobre todas as mensagens. Por esse motivo, modelos que possuem essas funções repetidas tem seu *fitness* penalizado.

A penalização utilizada é dinâmica, ou seja, são maiores nas primeiras gerações, diminuindo com a evolução dos ciclos. Essa estratégia foi adotada de forma a não penalizar de maneira exagerada bons indivíduos que evoluíram por diversos estágios.

4.5 Dicionários

A utilização de dicionários revela-se como uma característica muito importante, utilizada em grande parte dos trabalhos na literatura (como demonstrado nas Tabelas 3.2 e 3.3 do Capítulo 3), uma vez que os modelos dependem desses recursos para buscar as polaridades das palavras para avaliação [11, 180, 181]. Em [2] o autor afirma, inclusive, que a utilização de léxicos é a *feature* mais importante para incrementar a capacidade de predição dos classificadores. O trabalho de [52] também reforça a importância dos dicionários, classificando-os como o recurso mais relevante para a maioria dos algoritmos de Análise de Sentimentos.

Em [11], o autor analisa a diferença entre trabalhos da área de Análise de Sentimentos com bons resultados e afirma que grande parte deles tem como diferencial a utilização de um conjunto de dicionários léxicos e *features* específicas para trabalhar com as polaridades das palavras fornecidas pelos mesmos. A hipótese levantada é que a combinação de diversos léxicos pode reverter-se em melhores resultados para as técnicas de AS.

A escolha dos dicionários utilizados nesta pesquisa foi feita com base nos trabalhos relacionados levantados no Capítulo 3. Foram selecionados os léxicos mais utilizados e com disponibilidade para *download* e utilização de forma gratuita. Os dicionários empregados e a quantidade de palavras contidas em cada um (divididas por classe de polaridade) podem ser vistos na Tabela 4.4.

Tabela 4.4: *Dicionários utilizados no trabalho*

	Palavras		
Dicionário	**Positivas**	**Negativas**	**Total**
LIU	2006	4801	6807
Sentiwordnet	15439	16908	32347
AFINN	877	1599	2476
Vader	3300	4143	7443
Slang	15298	48827	64125
Effect	3298	2427	5725
SemEval2015	600	330	930
NRC	2312	3324	5636
General Inquirer	1914	2292	4206
Sentiment140	38312	24336	62648
MPQA SL	2718	4912	7630
TODOS	67752	90591	158343

Uma característica dos dicionários utilizados é a escala empregada para a definição da polaridade das palavras. Como é possível observar na Tabela 4.5, nem todos os léxicos utilizam a mesma grandeza de valores, entretanto, em todos eles é possível identificar o rótulo ao qual um valor pertence.

Tabela 4.5: *Saída dos dicionários utilizados no trabalho*

Dicionário	**Valores do dicionário**
LIU	positivo/negativo
Sentiwordnet	$[-1, +1] \in \mathbb{R}$
AFINN	$[-5, +5] \in \mathbb{Z}$
Vader	$[-4, +4] \in \mathbb{Z}$
Slang	$[-2, +2] \in \mathbb{Z}$
Effect	positivo/negativo
SemEval2015	$]-1, +1[\in \mathbb{R}$
NRC	$[-1, +1] \in \mathbb{Z}$
General Inquirer	$[-1, +1] \in \mathbb{Z}$
Sentiment140	$[-5, +5] \in \mathbb{R}$
MPQA SL	$[-1, +1] \in \mathbb{Z}$

Para a utilização neste trabalho, é levado em consideração a classe da palavra e não sua intensidade, ou seja, a saída dos dicionários são convertidas para um valor s, sendo que $s = [-1, 1] \in \mathbb{Z}$, com $s < 0$ representando a classe negativa e $s > 0$ positiva. Essa estratégia pode ser vista em outros trabalhos da literatura que utilizam uma solução composta de vários léxicos com saídas diferentes [153].

Bing LIU

Originalmente denominado *Opinion Lexicon*, é amplamente referenciado como Dicionário de LIU, graças a um de seus autores. Contém aproximadamente 6.800 termos em inglês, organizados em um dicionário de palavras positivas e outro de palavras negativas.

O léxico vem sendo compilado desde a sua concepção, em 2004, quando foi publicado no artigo [79]. Os autores selecionaram 30 adjetivos com orientação positiva e negativa para utilização na base de construção do dicionário e, por meio do WordNet, procederam com a expansão do léxico para a quantidade de palavras atual.

O dicionário está disponível para *download* gratuitamente no site do projeto [3] e pode ser utilizado por outras pesquisas de AS.

Sentiwordnet

Sentiwordnet, descrito em detalhes em [50], é um dicionário léxico formado por um conjunto de palavras – chamadas de *synset* – contendo 3 valores para cada uma delas: um grau de positividade, negatividade e objetividade, sendo o último calculado como 1−(grau de positividade + grau de negatividade). Os valores para cada um dos atributos variam de 0 a 1 $\in \mathbb{R}$.

Está disponível em duas versões (1.0 e 3.0) para *download* no site oficial[4]. O presente trabalho faz uso da versão 3.0 do léxico.

AFINN

AFFIN é um dicionário de palavras em inglês avaliadas em uma escala de -5 (extremamente negativo) a 5 (extremamente positivo), representando sua orientação semântica. Contém 2.476 termos manualmente anotados por [134] entre 2009 e 2011 e está disponível gratuitamente para *download* no site oficial[5].

[3] https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html
[4] http://sentiwordnet.isti.cnr.it/
[5] http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=6010

Vader

O dicionário VADER (*Valence Aware Dictionary and sEntiment Reasoner*) [60] é um léxico amplamente utilizado na literatura e parte de um classificador de sentimentos que leva o mesmo nome. Foi empiricamente validado por especialistas humanos e é especialmente ajustado para contextos de análises em redes sociais.

As palavras são classificadas em uma faixa de -4 (extremamente negativo) a 4 (extremamente positivo). O dicionário contém 7.443 termos e pode ser encontrado para *download* gratuitamente[6].

Slang

O dicionário Slang contém mais de 60 mil palavras e foi criado com o objetivo de ser utilizado em pesquisas de AS baseadas em redes sociais, como o *Twitter*. As palavras foram obtidas do UrbanDicionary[7] utilizando a funcionalidade de sinônimos fornecida pelo site. As polaridades possíveis para cada um dos termos do dicionário são: -2 (extremamente negativo), -1 (negativo), 0 (neutro), 1 (positivo) e 2 (extremamente positivo).

O trabalho de construção do dicionário e resultados de utilização na AS podem ser encontrados em [191]. O léxico pode ser encontrado no site do projeto[8].

Effect

O léxico Effect +/- contém uma série de palavras manualmente anotadas por especialistas quanto à sua polaridade. Além de um dicionário de termos positivos e negativos [33], fornece para *download*[9] uma série de outros recursos como léxico de argumentação, corpus para treinamento, entre outros.

O dicionário de palavras fornece 5.725 entradas, divididas nas classes positiva (3.298 palavras) e negativa (2.427 palavras).

SemEval 2015

Esse dicionário foi utilizado como um conjunto oficial de teste no evento SemEval 2015[10], *Task* 10, *Subtask* E. Os valores das polaridades das palavras variam de -1 a

[6] https://github.com/cjhutto/vaderSentiment
[7] https://www.urbandictionary.com/
[8] http://slangsd.com/
[9] https://mpqa.cs.pitt.edu/lexicons/effect_lexicon/
[10] alt.qcri.org/semeval2015/

$1 \in \mathbb{R}$, e está disponível para *download*[11] gratuitamente. Possui 930 termos, divididos em positivos (600) e negativos (330). É explanado em mais detalhes no trabalho publicado em [162].

NRC

O *NRC Lexicon* é formado por uma lista de palavras em inglês e suas polaridades (positiva e negativa) além da associação com oito emoções básicas – raiva, medo, antecipação, confiança, surpresa, tristeza, alegria e desgosto [125]. Muitas das anotações do léxico foram realizadas por meio de um trabalho de *crowdsourcing*.

O dicionário é livre para uso em pesquisa, e é disponibilizado para *download* na página do projeto[12], que contém mais informações sobre os trabalhos.

General Inquirer

O *General Inquirer* é um dos mais antigos dicionários utilizados atualmente [22]. Desenvolvido na década de 1960, publicado em [179], foi criado para a análise de conteúdo por cientistas sociais, políticos e psicólogos para avaliar características das mensagens a serem apreciadas.

Considerando somente as palavras classificadas como positiva e negativa, o dicionário possui 4.206 entradas (1.915 positivas e 2.291 negativas). Mais informações e os arquivos com os termos podem ser obtidos na página do projeto[13].

Sentiment 140

Criado a partir da rotulação automatizada de milhares de *tweets* com base na presença de *emoticons* positivos e negativos e a posterior verificação de frequência das palavras mais comuns em uma das duas classes [124], o *Sentiment140* fornece um conjunto de 62.648 unigramas (além de milhares de bigramas) e suas respectivas polaridades, variando de -5 a 5 [22, 61].

Disponível para uso acadêmico, pode ser adquirido no site[14] de forma gratuita.

[11] http://saifmohammad.com/WebDocs/lexiconstoreleaseonsclpage/SemEval2015-English-Twitter-Lexicon.zip

[12] https://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm

[13] http://www.wjh.harvard.edu/~inquirer/

[14] http://saifmohammad.com/Lexicons/Sentiment140-Lexicon-v0.1.zip

MPQA Subjectivity Lexicon

O *MPQA (Multi-Perspective Question Answering) Subjectivity Lexicon* (MPQA SL) é um dicionário léxico criado a partir de diversas fontes. Algumas palavras foram selecionadas de recursos desenvolvidos de forma manual. Outras entradas foram capturadas automaticamente por meio de dados anotados e não anotados. Muitos dos termos foram coletados como parte da pesquisa publicada em [155].

Fornece 2.718 palavras positivas e 4.912 negativas, e mais detalhes sobre o dicionário podem ser encontrados em [189] e na página oficial[15].

4.6 Ponderação dos dicionários

Apesar da importância da utilização de dicionários para a classificação de sentimentos [11, 104, 22, 2], a simples inclusão de mais léxicos na solução proposta não necessariamente resulta na melhoria direta dos resultados gerais dos classificadores. Dependendo da forma como o modelo atribui polaridade às mensagens, os resultados podem, inclusive, piorar. Um exemplo é o aumento na quantidade de mensagens neutras classificadas como positivas ou negativas, resultante de mais palavras disponíveis para a utilização pelos classificadores.

Tabela 4.6: *Dicionários utilizados no trabalho e seus atributos de peso*

Dicionário	Atributo
LIU	w_1
Sentiwordnet	w_2
AFINN	w_3
Vader	w_4
Slang	w_5
Effect	w_6
SemEval2015	w_7
NRC	w_8
General Inquirer	w_9
Sentiment140	w_{10}
MPQA SL	w_{11}

[15] `https://mpqa.cs.pitt.edu/lexicons/subj_lexicon/`

De forma a representar a importância dos dicionários utilizados na pesquisa, foi incluído um atributo que representa o peso de cada um dos léxicos na avaliação das mensagens. Essas propriedades são utilizadas pelas funções da PG, em especial pelo método 3 da Tabela 4.1 (*polSumAVGW*). Detalhes dessa configuração podem ser vistos na Tabela 4.6.

A intenção é que a própria PG possa determinar a importância de cada léxico para a avaliação das mensagens no domínio do problema, inclusive escolhendo ignorar o dicionário para o processo – definindo, nesse caso, seu peso como zero. Os pesos dos dicionários na PG são determinados, principalmente, pelo terminal δ, que varia entre -2 e 2 com $\delta \in \mathbb{R}$, conforme apresentado na equação 4-1.

Uma vez que a ponderação dos léxicos será definida pela PG, a sua variação está intimamente ligada ao operador genético de mutação. Como a probabilidade desse operador ser escolhido é de apenas 5% (como demonstrado na Tabela 4.3), a chance de um dos nós terminais que representa o peso de um dicionário passar por mutação é muito baixa, e fica muito dependente dos valores aleatórios gerados na criação do indivíduo.

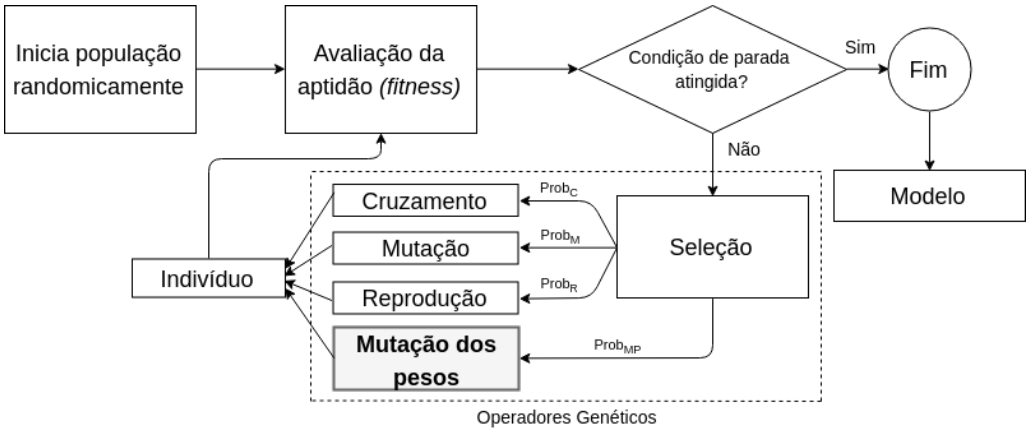

Figura 4.3: *Fluxo geral de funcionamento da Programação Genética com uma mutação específica para os pesos*

De forma a facilitar a modificação desses pesos durante as gerações para a criação de possíveis indivíduos mais aptos, foi estabelecido um operador de mutação especial para os valores dos terminais presentes na árvore – no caso deste trabalho, especificamente o terminal δ. Esse processo acontece de forma independente da mutação original da PG e, inclusive, podem ocorrer simultaneamente. A Figura 4.3 ilustra o fluxo modificado da PG com a inclusão da mutação especial para os pesos. O atributo $Prob_{MP}$ representa a probabilidade de um indivíduo ser processado por esse operador, com valor fixado em 0,5 ou 50%.

4.7 Faixa de valores das classes

A avaliação das mensagens considerará 3 classes possíveis: positiva, negativa ou neutra. Sendo r o valor resultante do processo de avaliação dos *tweets* pelos modelos m gerados pela PG, tem-se que $r \in \mathbb{R}$. Como o objetivo do trabalho é a classificação das entradas, é necessário que sejam definidos valores limite para cada uma das classes.

Tendo em vista a abordagem de utilização de dicionários léxicos e a saída da PG, uma alternativa possível seria considerar o valor zero para a classe neutra, valores maiores que zero para a classe positiva e menores que zero para a classe negativa, conforme demonstrado na Equação 4-12 (m representa um modelo de classificação $m \in M$ que retorna um valor $r \in \mathbb{R}$ de avaliação para um *tweet* $t \in T$). Apesar de simples, essa forma de avaliação de sentimentos é utilizada em diversos trabalhos de pesquisa [46, 127, 89, 3, 130, 180].

$$r = \begin{cases} positivo, & m(t) > 0 \\ neutro, & m(t) = 0 \\ negativo, & m(t) < 0 \end{cases} \tag{4-12}$$

A principal limitação dessa abordagem tem relação com a classificação das mensagens neutras. Considerando a fórmula exposta acima para a avaliação, uma mensagem seria classificada como neutra somente em casos em que o modelo retornasse o valor zero. Tendo em vista que a abordagem proposta neste trabalho combina uma série de dicionários, é razoável supor que muitas palavras presentes nos *tweets* serão encontradas, atribuindo, portanto, polaridade para a mensagem.

Haja vista o exposto acima, a definição dos limites de valores para cada uma das classes foi incluída no escopo de responsabilidades da própria PG, ou seja, fica a cargo da técnica definir as fronteiras que mais se adequam a cada categoria de mensagem. Note que, com isso, é possível ainda que a PG identifique que a melhor alternativa é manter a classe neutra com o valor zero. De todo modo, o contrário não é verdadeiro, ou seja, é preferível que a PG confirme que essa forma de atribuir polaridade é a mais adequada do que não poder alterar os limites caso necessário.

Dado o exposto, a faixa de valores para cada uma das classes é definida pela própria PG, por meio da função *neutralRange* (função 13 da Tabela 4.1). Esse método define os valores inferior e superior para a classe neutra. Com isso, considerando o mesmo conjunto M de modelos m, cada *tweet* $t \in T$ será classificado conforme a regra apresentada

na Equação 4-13.

$$r = \begin{cases} positivo, & m(t) > SR \\ neutro, & IR \leq m(t) \leq SR \\ negativo, & m(t) < IR \end{cases} \tag{4-13}$$

Sendo o Limite Superior do valor neutro identificado por SR e o Limite Inferior por IR, $IR, SR \in \mathbb{R}$. A Figura 4.4 apresenta de forma visual o esquema proposto para as fronteiras de valores de cada uma das classes.

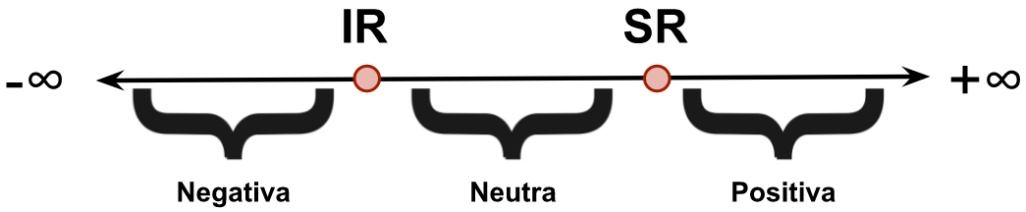

Figura 4.4: *Limites dos valores para cada uma das classes disponíveis: positiva, negativa e neutra*

CAPÍTULO 5

Experimentos

Este Capítulo apresenta os experimentos realizados no trabalho, com o objetivo de embasar o leitor com as informações necessárias para o entendimento dos resultados obtidos, apresentados no Capítulo 6. As experimentações foram conduzidas usando a linguagem de programação *Python*[1] na versão 3. Todas as bibliotecas utilizadas, apresentadas na Seção 5.2, foram escolhidas com a premissa de serem compatíveis com a linguagem supracitada.

Para cada um dos cenários foram executados 30 ciclos de treinamento, gerando, por sua vez, 30 modelos de classificação. Por conta da natureza estocástica presente na Programação Genética, esse processo busca garantir que os resultados dos modelos não derivem somente de processos puramente aleatórios. Para a apresentação e comparação dos resultados com outras técnicas de classificação, serão considerados os melhores modelos de cada conjunto.

Em cada modelo, as métricas calculadas são a acurácia (Equação 4-10), precisão (Equação 4-8), revocação (Equação 4-9) e *F1-score* (Equação 4-7). Os valores são calculados para cada uma das classes (4-7) e a média entre elas. Essas são as métricas mais utilizadas em avaliação de classificadores. Além disso, são calculadas as médias e desvio padrão entre os resultados dos modelos. Para determinar, quando necessário, se as diferenças são estatisticamente significativas, será utilizado o teste t pareado (Equação 4-11), com $\alpha = 0.05$.

Os resultados obtidos serão comparados com pesquisas que fizeram uso da mesma base de dados analisada, de forma a identificar se a estratégia é competitiva. Além disso, os valores serão confrontados com o resultado de classificadores clássicos utilizados na literatura.

[1] `https://www.python.org/`

5.1 *Benchmark*

Como *benchmark* foi utilizada a base de mensagens fornecida pelo evento SemEval 2014 (*International Workshop on Semantic Evaluation*), uma das principais competições na área de Processamento de Linguagem Natural (PLN) [163]. O evento é dividido por tarefas (*Tasks*), que possuem objetivos distintos dentro da área de PLN. Para este trabalho, foi usada a base de dados da *Task 9b - Sentiment Analysis in Twitter* (Análise de Sentimentos em *Twitter*) [163]. São disponibilizadas bases de treinamento e de testes para *download*[2] no site do evento[3].

A base de treinamento utilizada no trabalho possui 9.684 mensagens – classificadas como positivas, negativas e neutras – e está organizada conforme apresentado na Tabela 5.1 e na Figura 5.1. Como é possível observar, a maior parte das entradas pertence à classe neutra (4586), seguida das mensagens positivas (3640) e, por fim, negativas (1458). Analisar a distribuição das classes do conjunto de dados utilizado no trabalho é importante, uma vez que pode influenciar no treinamento dos modelos.

Tabela 5.1: *Distribuição das mensagens de treinamento*

Polaridade	**Mensagens**	**% do total**
Positiva	3640	38%
Negativa	1458	15%
Neutra	4586	47%
TOTAL	9684	100%

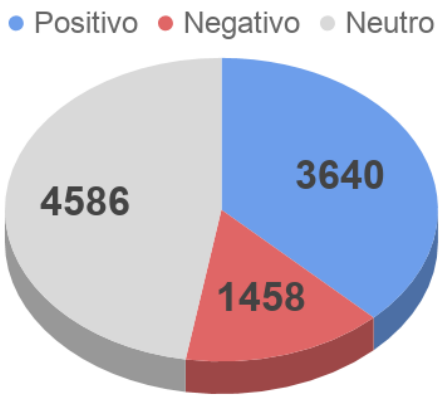

Figura 5.1: *Distribuição de polaridades das mensagens de treino*

[2] `http://alt.qcri.org/semeval2014/task9/`

[3] São disponibilizados os identificadores das mensagens, de modo que o usuário interessado deve fazer o *download* por meio de alguma ferramenta

O evento também disponibiliza uma base de teste com 8.987 mensagens, que serve como critério de avaliação e comparação entre os trabalhos submetidos para cada *Task*. Essa base, por sua vez, é dividida em 5 sub-bases, como apresentado na Tabela 5.2.

Tabela 5.2: *Distribuição das mensagens de teste*

Base	Mensagens	% Positiva	% Negativa	% Neutra
Twitter2013	3813	41%	16%	**43%**
Twitter2014	1853	**53%**	11%	36%
Sarcasmo	86	38%	**47%**	15%
SMS	2093	24%	19%	**57%**
LiveJournal	1142	**37%**	27%	36%
TOTAL	8987	39%	17%	**44%**

Assim como foi feito para o conjunto de treinamento, é importante investigar a distribuição das classes nas mensagens de teste. O resultado dessa análise para cada uma das 5 sub-bases é apresentado na Figura 5.2, bem como na própria Tabela anterior. É interessante notar que a classe neutra é dominante em 2 dos 5 subconjuntos e, em alguns casos, como ocorre na base SMS, é significativamente maior que a segunda maior classe observada.

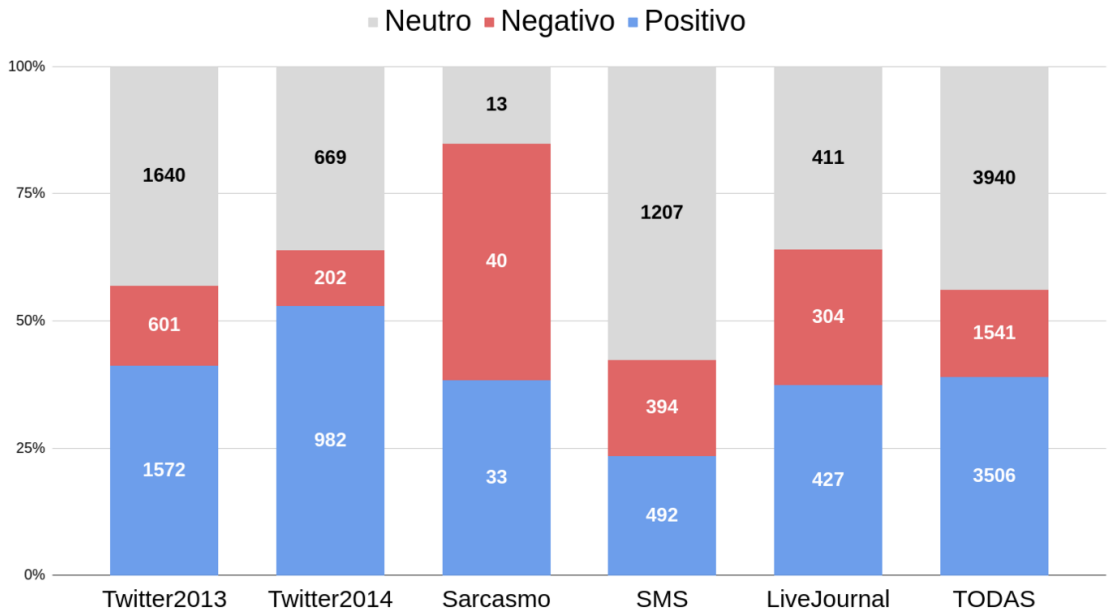

Figura 5.2: *Distribuição de polaridades das mensagens de teste*

Para a criação do *ranking* dos trabalhos submetidos, leva-se em consideração o *F1-score* médio das mensagens positivas e negativas, de acordo com a Equação 4-5. Note

que, apesar de considerar o *F1-score* médio somente das classes positiva e negativa, o classificador não é binário [163]. Ainda, vale a pena destacar que a classe negativa domina somente na base de Sarcasmo, justamente a que possui a menor quantidade de mensagens, refletindo diretamente na sua participação total na base de teste, representando somente 17% das entradas.

Como é possível perceber, as bases de treinamento e teste são desbalanceadas, ou seja, há uma classe que domina sobre as outras na totalidade das mensagens. Isso justifica a utilização do *F1-score* como métrica dos trabalhos pois, assim como discutido na Seção 4.2, outra medida pode não refletir de forma adequada a qualidade de predição do classificador.

5.2 Bibliotecas de apoio

Para apoiar o desenvolvimento de todos os operadores e recursos da PG foi utilizada a biblioteca DEAP[4] (*Distributed Evolutionary Algorithms in Python*), escrita na linguagem *Python* e disponível para uso gratuito. Fornece abstrações para a implementação de várias classes de Algoritmos Evolucionários, como Algoritmos Genéticos, Programação Genética, entre outros [57].

Especificamente para o contexto de Programação Genética, DEAP fornece funcionalidades para controle de criacão das estruturas de árvores, operadores genéticos, parametrização das operações, *logs*, entre outras. Além disso, fornece um módulo de Programação Genética Fortemente Tipada[5] (discutida no Capítulo 2), utilizado neste trabalho.

Para o *stem* das frases – processo de redução de palavras flexionadas para sua forma raíz – foi utilizada a biblioteca *Stemming1.0*[6] do *Python*. A lista de *stopwords* – palavras que podem ser consideradas irrelevantes para a análise do texto – foi criada por meio da biblioteca NLTK (*Natural Language Toolkit*), disponível para *download* no site do projeto[7].

O treinamento e a criação dos modelos das principais técnicas de AM foram feitos com a utilização da biblioteca *Scikit-learn*[8], de código aberto, escrita em linguagem *Python* e que fornece uma série facilidades para os principais algoritmos de Aprendizado de Máquina [143].

[4] https://github.com/DEAP/deap
[5] https://deap.readthedocs.io/en/master/api/gp.html#deap.gp.PrimitiveSetTyped
[6] https://pypi.org/project/stemming/1.0/
[7] https://www.nltk.org/
[8] http://scikit-learn.org/

5.3 Comparação dos resultados

Os resultados obtidos com a abordagem utilizando a PG serão confrontados com outras pesquisas que fizeram uso do mesmo *benchmark*. O evento SemEval 2014 fornece um *ranking*[9] com o resultado de todos os trabalhos submetidos, separados por base de teste, o que facilita a comparação. Além disso, os resultados serão comparados com os valores de outras técnicas clássicas de Aprendizagem de Máquina, por meio do treinamento e teste de alguns classificadores com o apoio da biblioteca *Scikit-learn*. Os algoritmos utilizados foram *Support Vector Machine* (SVM) com kernel RBF (*Radial basis function*), *Naïve Bayes* (NB), Regressão Logística (RL), *Random Forest* (RF) e *Stochastic Gradient Descent* (SGD) otimizando um SVM linear.

Foge do escopo deste trabalho adentrar em detalhes sobre o funcionamento das técnicas supracitadas. Entretanto, por serem amplamente aplicadas em problemas de classificação, há diversos trabalhos disponíveis na literatura sobre os assuntos. O leitor interessado pode encontrar detalhes sobre SVM em [35, 178], *Naïve Bayes* [131, 97], Regressão Logística [77], *Random Forest* [27] e *Stochastic Gradient Descent* [25].

A biblioteca *Scikit-learn* fornece pacotes para apoiar no desenvolvimento de cada um dos algoritmos selecionados (SVM[10], NB[11], RL[12], RF[13] e SGD[14]), facilitando a implementação dos módulos e o reuso das *features*.

A escolha das técnicas foi motivada pela análise dos trabalhos relacionados e os respectivos algoritmos utilizados (apresentado na Tabela 3.1). Para todas elas, o mesmo conjunto de atributos foi utilizado. Além disso, não foram feitas modificações nos parâmetros dos algoritmos, ou seja, foram utilizadas as configurações padrão definidas pelos pacotes da biblioteca de apoio. As *features* selecionadas para a utilização foram:

- Léxicos: utilização dos mesmos dicionários empregados na PG, apresentados na Tabela 4.4, para o fornecimento das polaridades de cada termo da mensagem;
- Negação e intensificação: verifica se a mensagem possui termos que representem negação e intensificação, com o apoio de um dicionário contendo entradas representando cada uma dessas classes – o mesmo dicionário utilizado na PG;

[9] http://alt.qcri.org/semeval2014/task9/

[10] http://scikit-learn.org/stable/modules/svm.html

[11] http://scikit-learn.org/stable/modules/naive_bayes.html

[12] http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[13] http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[14] http://scikit-learn.org/stable/modules/sgd.html

- N-gramas: representação das palavras em unigramas, bigramas e trigramas, de modo a identificar, além das entradas unitárias, formas compostas nas mensagens avaliadas;
- Caixa alta: contagem de termos escritos completamente em caixa alta;
- Pontuação: identificação de pontuação na mensagem, como exclamação e interrogação;
- Palavras alongadas: verifica se a mensagem possui termos que contém caracteres repetidos;
- Stemização: redução de termos flexionados para sua forma raíz;
- *Part-of-speech*: Contagem de cada *PoS tag* contidos na mensagem.

Da mesma forma que os algoritmos, a escolha dos atributos foi motivada pela análise dos trabalhos relacionados. O objetivo foi aplicar a maior parte das características em comum usadas pelas principais pesquisas, conforme apresentado em detalhes na Tabela 3.3 do Capítulo 3. Detalhes sobre cada uma das *features* empregadas e as justificativas de utilização podem ser encontrados na Seção 2.1.1.

5.4 Fluxo geral da solução

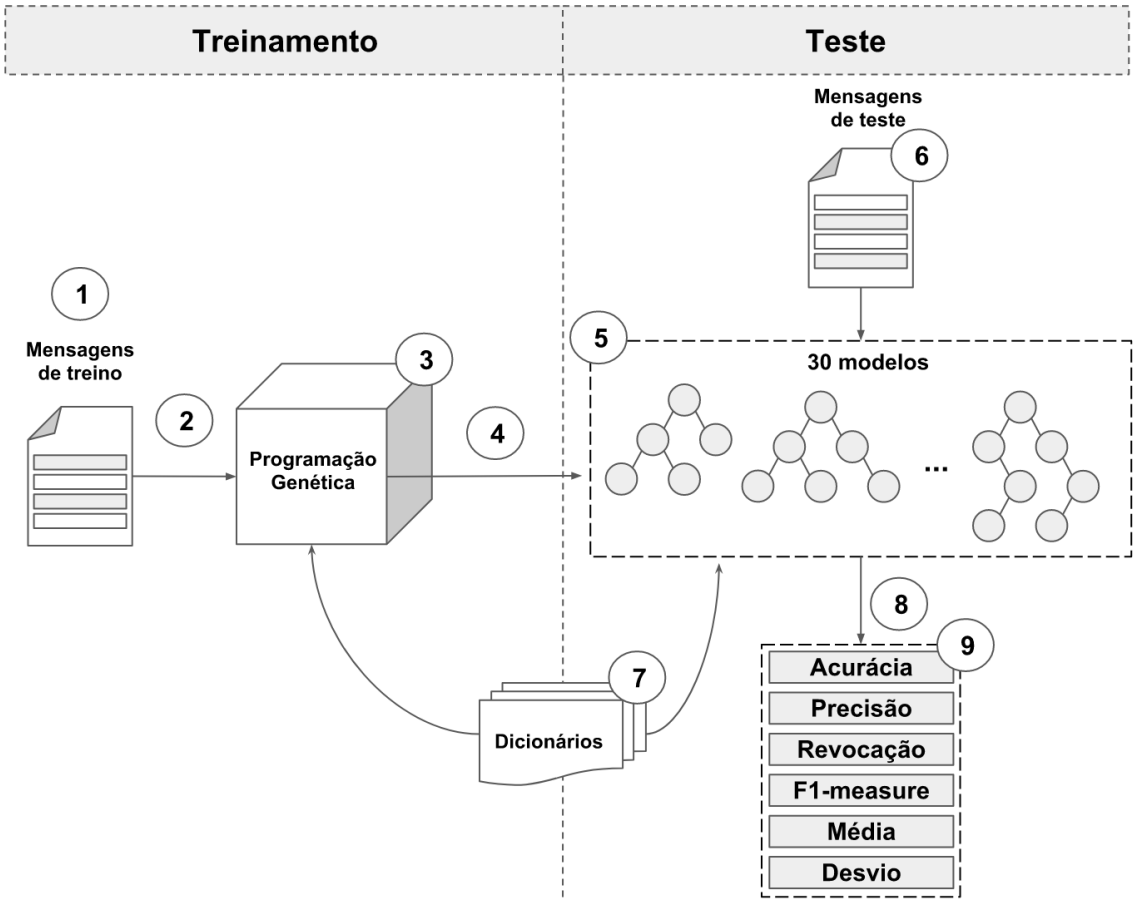

Figura 5.3: *Fluxo geral da solução proposta no trabalho*

De forma a facilitar a visualização da arquitetura geral da solução proposta e o entendimento da metodologia aplicada ao trabalho, a Figura 5.3 dá uma visão de alto nível das fases do processo e suas interfaces.

As mensagens de treinamento (item 1) do *benchmark* são submetidas como entrada para o processo de Aprendizado de Máquina Supervisionado fazendo uso da técnica de Programação Genética (itens 2 e 3). Vale lembrar que essas mensagens são enviadas em seu formato original, deixando a cargo da PG escolher as modificações mais relevantes. Esse processo de treinamento (item 3) é realizado de forma iterativa, até que as condições de parada sejam atingidas.

Para garantir resultados consolidados, são gerados 30 modelos para cada cenário estabelecido (itens 4 e 5). Para os processos de treinamento e teste são utilizados 11 dicionários léxicos (item 7), que contém um conjunto de palavras e suas polaridades (como apresentado na Tabela 4.4). As mensagens de teste do *benchmark* (item 6) são processadas de acordo com as regras definidas pelos modelos. As entradas classificadas,

então, são submetidas para avaliação (item 8) conforme algumas métricas de qualidade (item 9), sendo a principal delas o *F1-score* das classes positiva e negativa.

CAPÍTULO 6

Resultados

Neste Capítulo são apresentados os resultados da pesquisa, considerando os valores obtidos com a Programação Genética e com outras técnicas clássicas de Aprendizado de Máquina – *Support Vector Machine* (SVM), *Naïve Bayes* (NB), Regressão Logística (RL), *Random Forest* (RF) e *Stochastic Gradient Descent* (SGD) – conforme estratégia definida no Capítulo 4. Além disso, os resultados alcançados são confrontados com outros trabalhos que empregam o mesmo *benchmark*.

Para cada um dos cenários da PG são treinados 30 modelos e calculados, além dos valores individuais de cada um deles, a média e os melhores resultados. Ademais, de forma a identificar a diferença entre os modelos gerados, o desvio padrão também é medido. Para definir se as diferenças entre os resultados são estatisticamente significativas, é utilizado o teste t pareado, com um nível de confiança fixado em 95%, como explanado em detalhes na Seção 4.2. O melhor modelo de cada cenário é utilizado para a avaliação das mensagens de teste.

Os primeiros resultados dos testes realizados com a solução proposta são demonstrados na Tabela 6.1, que apresenta a média das principais métricas calculadas, organizadas por base. Os maiores valores de cada métrica estão em negrito e os piores resultados estão sublinhados para facilitar a leitura. F1 P/U/N representa o *F1-score* médio das classes positiva, neutra e negativa e F1 P/N refere-se à mesma métrica sem a classe neutra.

É possível notar que o *F1-score* médio das 3 classes (F1 P/U/N) é menor (em 4 de 5 bases) que a mesma métrica considerando a média das classes positiva e negativa (F1 P/N). Isso se justifica pelo fato da última ser o cálculo oficial utilizado pelo *benchmark* para a comparação dos trabalhos e, consequentemente, a métrica empregada como o *fitness* da PG, ou seja, o valor que o modelo busca maximizar. Esse resultado demonstra que a técnica é eficaz na busca por modelos que potencializem a função objetivo definida.

Observa-se, também, que os valores médios da acurácia são maiores que o *F1-score* das classes positiva e negativa em 3 bases (Twitter2013, Twitter2014 e SMS), além de apresentar resultados superiores quando considerada a avaliação de todas as mensagens (última linha da Tabela). Entretanto, conforme explanado na Seção 4.2, para conjuntos de

Tabela 6.1: *Resultados das principais métricas do modelo criado pela PG*

		Média			
Base	**Acurácia**	**Precisão**	**Revocação**	**F1 P/U/N**	**F1 P/N**
Twitter2013	64.38	61.85 ±0.08	64.41 ±0.07	62.66 ±0.06	62.78 ±0.07
Twitter2014	63.19	58.22 ±0.13	60.05 ±0.05	58.16 ±0.09	61.31 ±0.11
Sarcasmo	46.51	51.11 ±0.13	54.71 ±0.20	44.25 ±0.08	48.38 ±0.09
SMS	**68.99**	65.06 ±0.15	66.36 ±0.02	65.29 ±0.06	60.38 ±0.01
LiveJournal	67.16	**67.42** ±0.06	**66.74** ±0.04	**67.01** ±0.02	**68.53** ±0.03
TODAS	66.03	63.94 ±0.09	63.82 ±0.04	63.83 ±0.05	62.33 ±0.06

dados desbalanceados, como o utilizado neste trabalho (Figura 5.2), a acurácia pode não representar de forma justa a qualidade de um classificador.

As bases Twitter2013 e Twitter2014 apresentam maiores valores de *F1-score* considerando a média somente das mensagens positivas e negativas (última coluna), em detrimento da versão da métrica com a classe neutra incluída (penúltima coluna). Esse comportamento é desejado, uma vez que o objetivo do trabalho é maximizar o F1 P/N.

A base LiveJournal obteve os melhores resultados em 4 das 5 métricas (exceto acurácia). Esse conjunto contém mensagens com menor incidência de gírias e palavras informais reforçando, portanto, a dificuldade em lidar com textos que apresentam linguagem coloquial, característica das redes sociais. Essa base apresentou os resultados mais estáveis quando consideradas todas as métricas.

Tendo em vista somente a acurácia das classificações, o melhor resultado é observado na base de SMS. Isso mais uma vez demonstra que, tão importante quanto a quantidade de mensagens avaliadas corretamente, é a distribuição dessas avaliações entre as classes disponíveis. É importante lembrar, ainda, que essa base possui a maior disparidade em relação à distribuição de classes entre as mensagens, conforme foi demonstrado na Seção 5.1, sendo composta por 57% de entradas neutras.

Por fim, ao comparar o *F1-score* das 3 classes (F1 P/U/N) com a mesma métrica desconsiderando a classe neutra (F1 P/N), é possível perceber que a base de SMS possui a maior diferença nos resultados, apresentando um decréscimo de 8% da primeira para a segunda medida.

O menor resultado, para todas as métricas, foi obtido na base que contém entradas sarcásticas (Sarcasmo), o que atesta a dificuldade em tratar mensagens com essa figura de linguagem, sendo considerada um dos desafios abertos da área de Análise de Sentimentos [149, 106].

Importante destacar ainda que, no estudo publicado em [63], o autor reporta que a habilidade de detecção de sarcarmo por humanos atingiu uma acurácia de apenas 62.59%,

muito próxima do valor resultante dos melhores classificadores do *benchmark*. Por fim, cabe salientar que há um nível ainda maior de dificuldade ao incluir a classe neutra na detecção e classificação de mensagens sarcásticas. Como exemplo, a Tabela 6.2 apresenta entradas do subconjunto de sarcasmo e as classes anotadas pelo *benchmark*.

Tabela 6.2: *Exemplo de mensagens de sarcasmo e suas classes*

Mensagem	**Classe**
what a beautiful saturday night with my pretty clothes	Positiva
John Fox kicking the field goal on 4th and 1 #shocked	Neutra
Not stoked at all for youth group tomorrow	Negativa

Como é possível perceber, não é trivial, mesmo para humanos, avaliar esse tipo de mensagem, o que torna essa tarefa um dos principais desafios da área de Análise de Sentimentos, com pesquisas dedicando-se exclusivamente ao tema [149, 119, 113, 106].

Tabela 6.3: *Comparação de resultados da PG com os trabalhos submetidos para SemEval 2014 (F1-score)*

Base	**Resultado PG**	**Majority Baseline**	**Top 3 SemEval**
Twitter2013	62.78	29.2	1º 72.12 2º 70.75 3º 70.40
Twitter2014	61.31	34.6	1º 70.96 2º 70.14 3º 69.95
Sarcasmo	48.38	27.7	1º 58.16 2º 57.26 3º 56.50
SMS	60.38	19	1º 70.28 2º 67.68 3º 67.51
LiveJournal	68.53	27.2	1º 74.84 2º 74.46 3º 73.99

A comparação dos resultados obtidos com a PG em relação aos trabalhos submetidos para o SemEval 2014 é apresentada na Tabela 6.3. Pode-se perceber, de

início, que todos os resultados foram superiores ao *Majority Baseline* – que representa um classificador que considera acertar somente a classe dominante da base de testes. Ao todo, 50 trabalhos foram enviados para a avaliação do *benchmark*.

É importante lembrar que, como demonstrado no Capítulo 4, para algumas bases de teste, a distribuição de mensagens entre as classes é consideravelmente desigual, o que poderia levar à construção de classificadores imprecisos, que priorizassem a avaliação de acordo com a base de prevalência, de forma a obter melhores resultados.

Além disso, percebe-se que, mesmo a base Sarcasmo apresentando os piores resultados entre todas as outras, essa apresentou *F1-score* 9.78 inferior ao primeiro colocado do *benchmark*, reforçando a dificuldade em classificar esse tipo de mensagem.

As bases Twitter2013 e Twitter2014 tiveram um *F1-score* de 62.78 e 61.31, respectivamente. Esses valores são 33.58 e 26.71 pontos maiores que o *Majority Baseline* de cada uma das bases, o que mostra que não há uma busca para acertos de mensagens somente da classe dominante da base de testes. Além disso, ao considerar os maiores valores dos trabalhos submetidos, nota-se uma diferença de 9.34 pontos para Twitter2013 e de 9.65 pontos para Twitter2014, o que dá indícios de que a técnica é competitiva com os outros trabalhos do *benchmark*.

O melhor resultado obtido pela PG foi alcançado na base LiveJournal, com um *F1-score* 68.53, ficando 41.33 pontos acima do *Majority Baseline* e apenas 6.31 pontos – ou aproximadamente 8% – menor que o melhor colocado no *benchmark*, reforçando a característica competitiva da abordagem.

A base de SMS, assim com as supracitadas, obteve um resultado consideravelmente maior que o *Majority Baseline*: 41.38. Ao mesmo tempo, pode-se perceber que, em comparação ao melhor resultado do *benchmark* na base, os valores da PG foram 14% menores.

Técnicas clássicas de Aprendizado de Máquina comumente empregadas na AST foram utilizadas para a criação de classificadores, de modo a permitir a comparação com os resultados retornados pela PG. Todos os modelos foram gerados com o apoio da biblioteca *Scikit-learn* e foi mantida a parametrização padrão de cada um dos algoritmos.

As *features* utilizadas tiveram como base os trabalhos relacionados e são descritas na Seção 5.3. Os valores alcançados por cada algoritmo, organizados por base de teste, são demonstrados na Tabela 6.4. A última coluna da Tabela apresenta os resultados da PG, de modo a facilitar a comparação[1].

[1] O sinal de asterisco representa que a diferença entre as médias da técnica e da PG são estatisticamente significativas, considerando o t teste com confiança de 95%

Tabela 6.4: *Principais resultados das técnicas (F1-score) utilizadas e a relação com a PG*

	F1-score médio classes pos e neg					
Base	**RF***	**SVM**	**NB***	**RL***	**SGD**	**PG**
Twitter2013	49.66	**63.75**	52.84	61.57	58.5	62.78
Twitter2014	47.5	**63.36**	51.24	60.48	61.24	61.31
Sarcasmo	43.23	41.38	**54.25**	47.26	35.07	48.38
SMS	40.89	62.56	43.1	51.32	**65.25**	60.38
LiveJournal	51.46	**69.94**	54.61	60.94	68.05	68.53
TODAS	48.73	**64.62**	51.2	59.32	62.53	62.33

Como é possível visualizar, a PG superou 4 das 5 técnicas utilizadas como parâmetro de comparação – RF, NB, RL e SGD – nas bases Twitter2013, Twitter2014, Sarcasmo e LiveJournal. A única exceção ocorreu na base de SMS, em que as técnicas SVM e SGD superaram os resultados obtidos pela PG. É importante destacar que foi justamente nessa base que a PG obteve a maior acurácia, como pode ser visto na Tabela 6.1.

Ao comparar os resultados da PG especificamente com o SVM, é possível observar que a última possui resultados superiores em 4 das 5 bases de teste, porém com a maior diferença sendo apenas de 2.18 pontos (3%), observada na base de SMS. Ainda, cabe frisar que a PG supera o classificador SVM na base de Sarcasmo em 7 pontos, resultado 17% superior.

A utilização do teste t pareado (apresentado na Equação 4-11) demonstra que a diferença de resultados entre a PG e o SVM não é estatisticamente significativa, ou seja, não há evidências para rejeitar a Hipótese nula $H_0 : \mu_1 = \mu_2$, que afirma a média das populações analisadas são iguais.

Além da comparação com os resultados obtidos por SVM, ao confrontar os valores da PG com as outras técnicas, percebe-se que, com exceção de SGD, as diferenças entre os algoritmos são estatisticamente significantes, considerando o nível de confiança fixado em 95%.

É importante destacar os resultados das abordagens na avaliação das mensagens da base de Sarcasmo. Com exceção do algoritmo *Naïve Bayes*, todos produziram um *F1-score* abaixo de 50, sendo o resultado da PG o segundo melhor. O SVM, melhor classificador no geral, atingiu o segundo pior resultado nessa base, com *F1-score* de 41.38, sendo superior somente aos resultados do SGD. Esses valores enfatizam a dificuldade de classificação desse tipo de mensagem.

Os resultados apresentados indicam que a Hipótese 1, levantada no Capítulo 1, é

válida, uma vez que os valores dos classificadores gerados por meio da PG mostraram-se competitivos – e muitas vezes superiores – com outros trabalhos da literatura e com as técnicas clássicas utilizadas na AST.

Uma forma de visualizar o comportamento geral do classificador e identificar os pontos críticos do modelo é por meio da matriz de confusão, que correlaciona as predições realizadas e as classes reais das mensagens (*Gold*). Diante disso, foi gerada a matriz para todas as mensagens avaliadas, apresentada na Tabela 6.5. A diagonal principal representa as mensagens classificadas corretamente pelo modelo.

Tabela 6.5: *Matriz de confusão do melhor modelo da PG para todas as mensagens*

	Gold Positivo	**Gold Negativo**	**Gold Neutro**
Pred Positivo	**2536**	266	949
Pred Negativo	240	**832**	425
Pred Neutro	730	443	**2566**

Como é possível observar nos resultados da diagonal principal, a maior parte das mensagens foi classificada corretamente. Pode-se perceber, também, que o maior problema encontrado no processo tem relação com as mensagens neutras sendo classificadas como positivas. Os falsos negativos, apesar de ocorrerem, são mais raros. Esse comportamento pode demonstrar que o modelo está atribuindo polaridade em excesso a algumas palavras e não está conseguindo adequar o intervalo de valores das classes da melhor forma. Além disso, os baixos valores atingidos na base de sarcasmo refletem diretamente no resultado geral do classificador.

Os resultados apresentados pela matriz de confusão podem servir de subsídio para futuras melhorias nos processos de classificação. Importante citar, entretanto, que não é trivial corrigir comportamentos inadequados de modelos de classificadores sem alterar práticas consideradas positivas, ou seja, realizar modificações que não prejudiquem a avaliação das mensagens preditas corretamente. Ademais, devido à métrica principal ser a *F1-score*, não basta somente aumentar a quantidade de acertos (acurácia) mas, sim, é necessário uma melhoria geral nos processos de predição do modelo.

6.1 Alteração no processo de treinamento da PG

Por conta da característica de funcionamento da Programação Genética e, principalmente, a forma como os indivíduos são avaliados e executados para identificar sua aptidão, a base de treinamento frequentemente é utilizada em sua totalidade, sem sua

divisão em subconjuntos [145, 93, 62]. Isso pode ter consequências diretas na generalização do modelo – a habilidade de um indivíduo obter bons resultados na análise de dados ainda não conhecidos [62] – podendo causar problemas como o sobreajuste – quando um modelo se ajusta demasiadamente ao conjunto de treinamento mas não apresenta bons resultados em outras avaliações de teste [98].

Com a intenção de melhorar a generalização dos modelos gerados pela PG, foi realizada uma alteração no fluxo geral da solução, apresentada anteriormente na Figura 5.3. Em vez de utilizar a totalidade das mensagens de treino para a criação de cada modelo, foram geradas $n = 10$ amostras com reposição e o indivíduo é avaliado para cada uma dessas partes. A hipótese levantada é a de que essa alteração no processo treinamento da PG pode beneficiar a generalização dos modelos e, consequentemente, melhorar os resultados gerais do classificador.

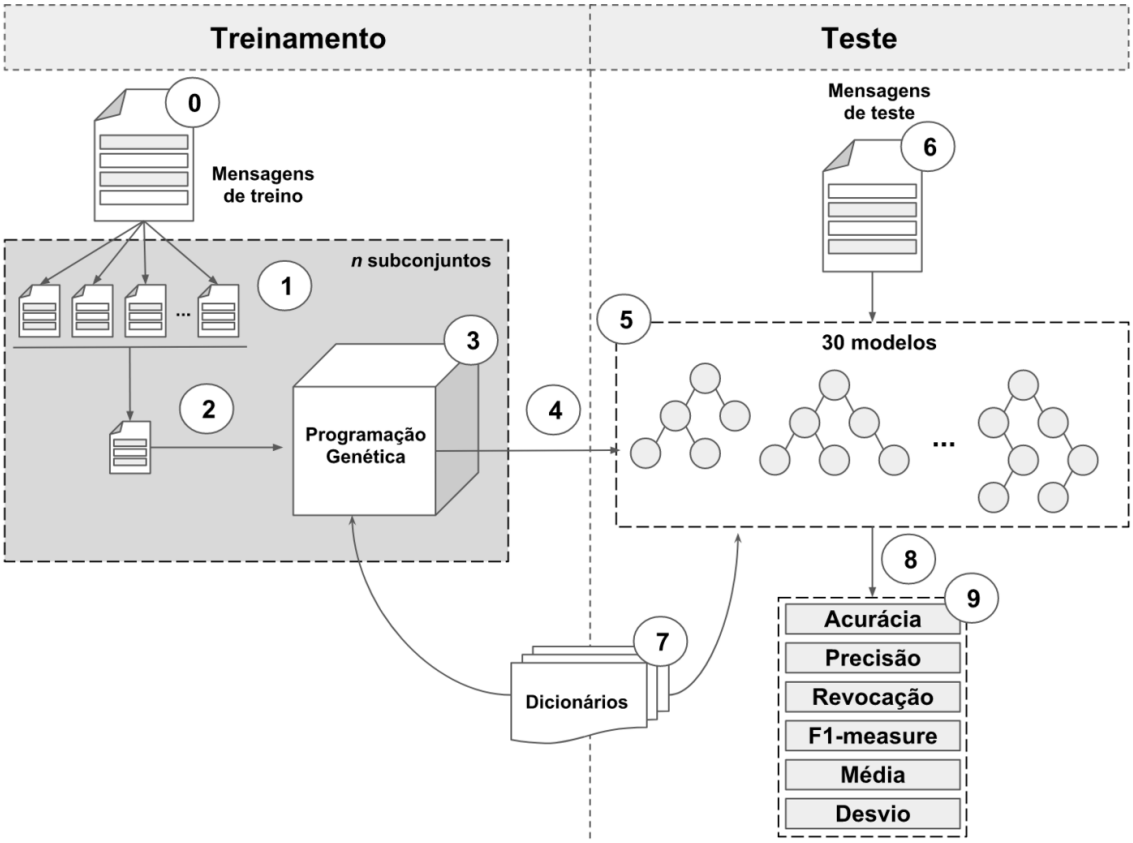

Figura 6.1: *Fluxo geral da solução modificado*

$$Fitness_{new} = \frac{\sum_{i=1}^{n} F1_i^{PN}}{n} \tag{6-1}$$

Ao alterar o processo de treinamento dos indivíduos da PG, a função de aptidão também é modificada. Em lugar da utilização do *F1-score* das classes positiva e negativa, conforme apresentado na Seção 4.2, a função *fitness* dos modelos foi alterada para a média aritmética do *F1-score* positivo e negativo das n partes da base de treino. A representação

matemática da nova função de aptidão é demonstrada na Equação 6-1, na qual *n* representa a quantidade de subconjuntos da base de treinamento e $F1_i^{PN}$ é o *F1-score* das classes positiva e negativa do *i*-ésimo subconjunto de treinamento. O fluxo atualizado com as modificações descritas é apresentado na Figura 6.1.

A principal mudança realizada no fluxo da solução tem relação com a divisão da base de treinamento em *n* partes, cada uma delas sendo avaliada e processada por um indivíduo da PG. A base original (item 0), portanto, é dividida em *n* parcelas aleatórias – por meio de reposição, em que as mensagens podem participar de mais de um subconjunto e também de nenhum deles – e cada uma dessas partes (item 1) passa pelo processo de treinamento da PG (itens 2 e 3). Note que cada uma das divisões do conjunto de treinamento é avaliada pelo mesmo modelo, ou seja, essa estratégia é diferente da apresentada na Figura 2.15 no Capítulo 2, ilustrando o funcionamento do *bagging*.

Após a implementação das alterações na estrutura geral da solução, novos ciclos de treinamentos foram realizados, com o objetivo de identificar possíveis alterações nos resultados dos classificadores gerados pela PG. Assim como foi feito no processo original, 30 novos modelos foram produzidos e as mesmas métricas apresentadas na Seção 4.2 foram aplicadas, com exceção do *fitness* modificado, exposto na Equação 6-1.

Os primeiros resultados obtidos após a alteração no processo de treinamento da PG são apresentados na Tabela 6.6. Assim como na versão anterior, os maiores valores de cada métrica estão em negrito e os piores resultados estão sublinhados para facilitar a leitura. F1 P/U/N representa o *F1-score* médio das classes positiva, neutra e negativa e F1 P/N refere-se à mesma métrica sem a classe neutra. Para facilitar a distinção entre as versões apresentadas, a configuração atualizada da PG será identificada como PG_a nas tabelas e textos que seguem.

Tabela 6.6: *Resultados das principais métricas do modelo criado pela PG atualizada (PG_a)*

		Média			
Base	**Acurácia**	**Precisão**	**Revocação**	**F1 P/U/N**	**F1 P/N**
Twitter2013	65.59	64.53 ±0.13	67.66 ±0.13	63.94 ±0.06	65.47 ±0.07
Twitter2014	63.25	58.28 ±0.14	64.12 ±0.13	59.06 ±0.10	62.31 ±0.10
Sarcasmo	48.84	49.43 ±0.09	49.8 ±0.20	46.31 ±0.09	48.04 ±0.10
SMS	65.6	64.26 ±0.17	**70.85** ±0.10	64.67 ±0.03	62.6 ±0.01
LiveJournal	**69.44**	**70.07** ±0.07	70.31 ±0.10	**69.18** ±0.03	**71.24** ±0.01
TODAS	65.44	64.71 ±0.12	67.8 ±0.11	64.39 ±0.05	65.5 ±0.06

Como é possível observar, houve um acréscimo nos valores das métricas em relação à versão anterior em todas as bases, com exceção de Sarcasmo. Mais uma vez, o *F1-score* médio das 3 classes é menor que a versão sem a classe neutra, o que mostra

que há uma convergência em direção ao *fitness*. A exceção continua sendo a base de SMS, que possui um F1 P/U/N maior que F1 P/N. A base LiveJournal possui as melhores métricas entre todos os conjuntos, com exceção da Revocação, o que mostra que essa base é consistente nos resultados obtidos.

A comparação com os trabalhos do *benchmark* – demonstrada na Tabela 6.7 – permite constatar que os resultados estão ainda mais próximos dos primeiros trabalhos, em especial a base LiveJournal, apenas 3 pontos menor que o melhor colocado (diferença de 4%). É importante destacar, ainda, que a diferença em relação ao *Majority Baseline* aumentou, o que demonstra o esforço da técnica em buscar um modelo generalista, atuando na predição de todas as classes.

Tabela 6.7: *Comparação de resultados da PG_a com os trabalhos submetidos para SemEval 2014 (F1-score)*

Base	Resultado PG_a	Majority Baseline	Top 3 SemEval
Twitter2013	65.47	29.2	1º 72.12 2º 70.75 3º 70.40
Twitter2014	62.31	34.6	1º 70.96 2º 70.14 3º 69.95
Sarcasmo	48.04	27.7	1º 58.16 2º 57.26 3º 56.50
SMS	62.6	19	1º 70.28 2º 67.68 3º 67.51
LiveJournal	71.24	27.2	1º 74.84 2º 74.46 3º 73.99

Do mesmo modo, é possível perceber que a base de Sarcasmo obteve resultados inferiores que a versão anterior, aumentando, consequentemente, a diferença para os melhores trabalhos do *benchmark*. Além da base de LiveJournal, destacada anteriormente, vale a pena evidenciar o crescimento das bases Twitter2013 e Twitter2014, a primeira delas pouco mais de 6 pontos abaixo do primeiro colocado e apenas 7% inferior aos valores das 3 primeiras pesquisas.

Da mesma forma como foi feito na versão original, é importante realizar a comparação dos resultados obtidos em relação às técnicas clássicas de AM. O *F1-score*

médio das classes positiva e negativa de cada um desses métodos e o resultado da PG atualizada (PG_a) são apresentados na Tabela 6.8.

Tabela 6.8: *Principais resultados das técnicas (F1-score) utilizadas e a relação com a PG_a*

	F1-score médio classes pos e neg					
Base	**RF***	**SVM**	**NB***	**RL***	**SGD**	***PG_a***
Twitter2013	49.66	63.75	52.84	61.57	58.5	**65.47**
Twitter2014	47.5	**63.36**	51.24	60.48	61.24	62.31
Sarcasmo	43.23	41.38	**54.25**	47.56	35.07	48.04
SMS	40.89	62.56	43.1	51.32	**65.25**	62.6
LiveJournal	51.46	69.94	54.61	60.94	68.05	**71.24**
TODAS	48.73	64.62	51.2	59.32	62.53	**65.5**

É possível perceber que os valores da PG_a superam 2 dos 5 métodos de AM em todas as bases – RF e RL. Cabe destacar, ainda, que os novos resultados são superiores em relação aos valores obtidos com o SVM em Twitter2013, Sarcasmo, SMS e LiveJournal. Ainda, a PG_a possui os melhores resultados entre todos os classificadores para os conjuntos de Twitter2013, LiveJournal e para todas as mensagens, o que reforça a característica competitiva da abordagem.

Ao analisar os resultados por meio do teste t pareado, não se pode afirmar que as diferenças entre os valores da PG_a e SVM são estatisticamente significativos, uma vez que o valor de $t = 1.6775$ e, consequentemente, $p = 0.1542$. Como se busca uma diferença que resulte em $p < 0.05$, não é possível descartar a Hipótese nula H_0, que afirma que $\mu_1 = \mu_2$. Apesar disso, a diferença entre os resultados da PG_a e os outros métodos apresentados são estatisticamente significativos, considerando a confiança de 95% adotada no presente trabalho.

Ao levar em consideração a avaliação de todas as mensagens de teste, a PG_a obtém os melhores valores, com um *F1-score* de 65.5. Isso demonstra que, apesar de não ser a melhor técnica para algumas bases específicas, a estratégia apresenta resultados consistentes no geral, resultando em um valor global superior às outras abordagens.

Os resultados revelam que a atualização na forma de treinamento da PG (PG_a) melhoraram o resultado geral do classificador em todas as bases de teste, com exceção de Sarcasmo, com uma perda de menos de 1%. Apesar da melhoria em alguns casos ser relativamente baixa, é importante considerar que não houve diferença significativa com relação ao custo necessário para o treinamento dos modelos entre as versões, ou seja, o

acréscimo nos resultados foi obtido sem o aumento no esforço computacional na fase de treinamento e teste. As bases Twitter2013 e Twitter2014 tiveram um crescimento de 4.3% e 1.6%, respectivamente.

Em LiveJournal, o aumento foi de aproximadamente 4%, o que fez com que a base, que já havia sido responsável pelos melhores resultados na configuração anterior, melhorasse ainda mais os valores obtidos. Considerando a avaliação de todas as mensagens do conjunto de teste, o crescimento obtido no *F1-score* foi de 5%.

A aplicação do teste t pareado mostra que as diferenças entre os resultados das versões são estatisticamente significativas, com um valor de $t = 3.519$ e $p = 0.017$, com 95% de confiança. Relevante destacar o aumento nos valores de variância e desvio padrão da PG_a, consequência da estratégia de treinamento adotada. Esses valores reforçam a hipótese de melhoria dos resultados gerais, levantados no início desta Seção, resultante da alteração no processo de treinamento, uma vez que é possível observar um incremento na capacidade de predição da PG.

A matriz de confusão dos resultados, apresentada na Tabela 6.9, dá uma visão geral do comportamento da solução. Como se pode perceber, há um aumento significativo na quantidade de palavras positivas e negativas preditas corretamente, em relação aos valores apresentados na matriz de confusão anterior (Tabela 6.5).

Tabela 6.9: *Matriz de confusão do melhor modelo da PG_a para todas as mensagens*

	Gold Positivo	**Gold Negativo**	**Gold Neutro**
Pred Positivo	**2689**	249	1122
Pred Negativo	380	**1156**	782
Pred Neutro	437	136	**2036**

A quantidade de mensagens neutras avaliadas corretamente, entretanto, diminuiu. Isso demonstra uma tendência da solução em buscar um maior *fitness*, uma vez que, apesar do classificador não ser binário, a métrica principal leva em consideração o *F1-score* médio somente das classes positiva e negativa [163]. Esse comportamento reflete diretamente nos valores retornados pelas métricas pois, como foi revelado na Tabela 6.6, o *F1-score* médio considerando a classe neutra é sempre menor que a mesma métrica somente das classes positiva e negativa (com exceção da base de SMS).

6.2 Combinação dos dicionários e limites das classes

Como discutido anteriormente, um dos benefícios da utilização da Programação Genética como técnica de Aprendizado de Máquina é a possibilidade de leitura e inter-

pretação dos modelos gerados. Desse modo, é possível analisar os resultados, de forma a entender o processo da solução [96].

Devido a maneira com que gera e avalia os modelos, a PG, de certa forma, realiza uma seleção de *features*, determinando as funções mais relevantes para o problema em que está sendo aplicada [66]. Com base no exposto, uma análise dos resultados obtidos com os experimentos pode evidenciar as funções e características consideradas mais significativas para a avaliação do *benchmark* pela PG.

Uma verificação da utilização dos dicionários incluídos na solução e a importância dada a eles pela PG (por meio dos pesos) pode revelar os léxicos mais adequados ao problema em questão, ou seja, os que forneceram maior benefício para o processo de avaliação do *benchmark*.

A Figura 6.2 apresenta os valores médios dos pesos de cada um dos dicionários para os dois cenários testados neste trabalho (PG e PG_a). Note que esse valor representa as médias dos pesos atribuídos para cada um dos léxicos nos 30 modelos gerados em cada versão.

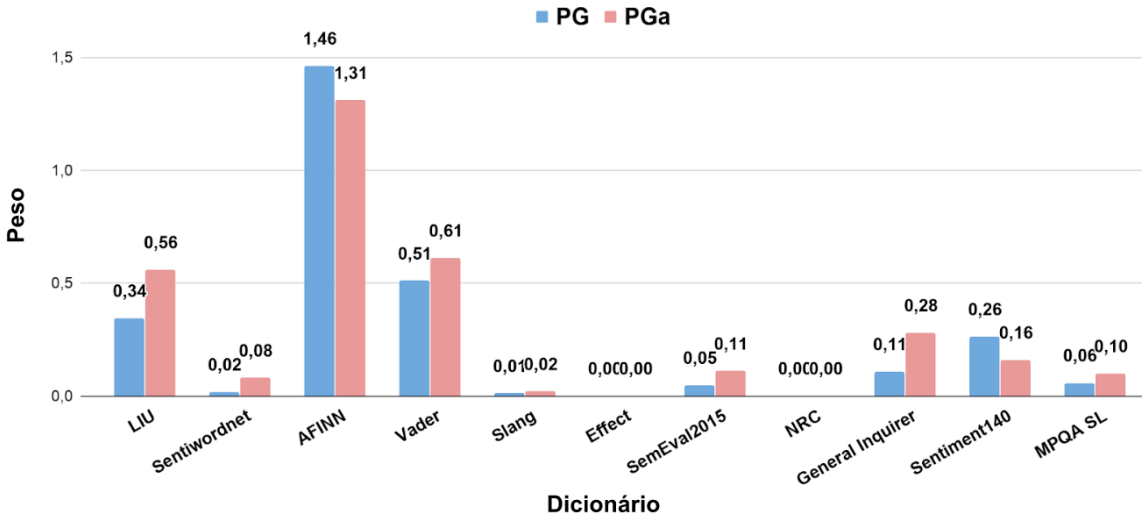

Figura 6.2: *Média dos pesos por dicionário utilizado nos modelos da PG e PG_a*

Como se pode observar, em ambos os tipos de PG considerados, a importância dos dicionários para a solução do *benchmark* manteve-se estável, ou seja, os que possuem maior peso são os mesmos. Além disso, os léxicos Effect (w_6) e NRC (w_8) receberam os menores valores em todos os modelos, com pesos abaixo de de 0.001. É importante salientar que a baixa relevância de um dicionário em determinado domínio não implica que o mesmo atribui polaridades incorretas às palavras, mas tão somente que o léxico não foi expressivo para determinada estratégia e *benchmark*. O dicionário AFINN (w_3) obteve o maior peso entre os léxicos utilizados, ou seja, a PG entendeu que ele auxilia de forma mais eficaz na maximização das predições dos modelos, seguido dos dicionários VADER (w_4) e LIU (w_1).

Assim como foi feito com relação aos valores dos pesos dos dicionários, as fronteiras de cada classe, definidas pela PG por meio da função *neutralRange*, foram levantadas. Essa característica é importante e permite a definição de limites dinâmicos nos valores de cada classe disponível, dependendo da combinação de funções e dicionários utilizados. A análise pormenorizada desses valores é, portanto, valiosa para entender como os modelos estão atribuindo polaridades para as mensagens.

A Figura 6.3 apresenta um gráfico que demonstra os valores mais utilizados para os limites inferior e superior da classe neutra. É possível perceber que o menor valor definido para o limite inferior foi pouco maior que -1. Da mesma forma, a maior medida para esse limite foi 4, desconsiderando o *outlier*[2]. Apesar disso, pode-se notar que mais de 75% dos valores atribuídos para o limite inferior são maiores que zero, ou seja, positivos.

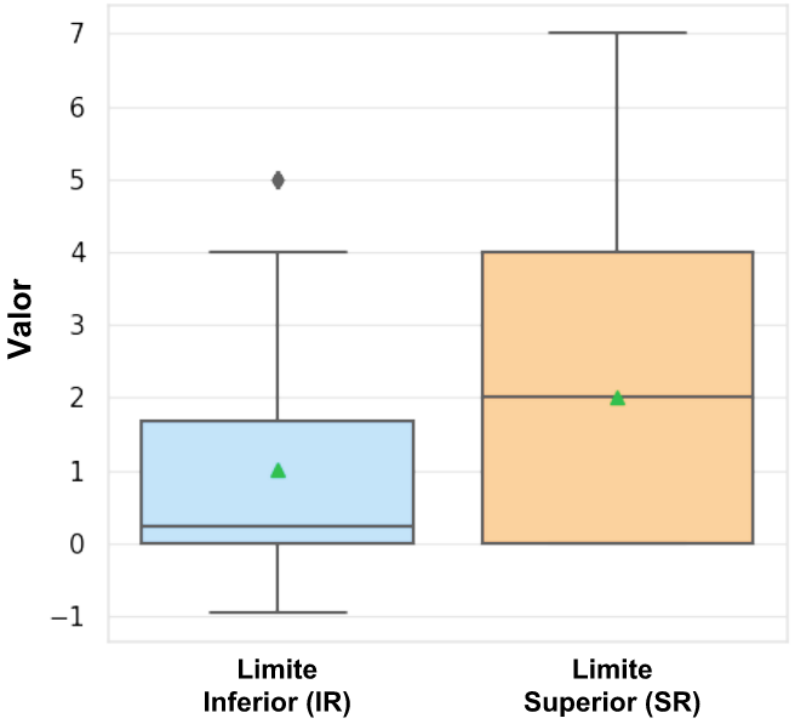

Figura 6.3: *Valores limite da classe neutra*

Do mesmo modo, é possível visualizar que o menor valor definido para o limite superior foi zero, ou seja, todas as entradas foram positivas e 75% delas estão entre 0 e 4, sendo o maior valor encontrado igual a 7. O valor médio para cada um dos limites é apresentado em detalhes na Tabela 6.10.

[2]Valor atípico encontrado, muito distante da maioria das entradas.

Tabela 6.10: *Valores médios dos limites da classe neutra definidas pela PG*

Média	
Limite inferior (IR)	**Limite superior (SR)**
1.0226	2.0080

Os valores apresentados na Tabela anterior demonstram uma adequação, por parte da PG, dos limites da classe neutra, com uma preferência para valores positivos. Essa adaptação pode estar sendo feita para balancear os resultados provenientes das avaliações feitas por meio dos dicionários. Isso demonstra, também, que a utilização da divisão clássica entre os valores dos limites, atribuindo a classe neutra somente para saídas igual a zero (como demonstrado em 4-12), não foi a configuração mais adequada encontrada pela PG.

É interessante, ainda, fazer uma análise do melhor modelo resultante do processo de evolução da Programação Genética. Por conta dos resultados superiores obtidos, será considerado o melhor indivíduo gerado na versão PG_a do experimento e, para facilitar a distinção, será referenciado como m_{best} nos textos que seguem. A Figura 6.4 representa os valores dos limites inferior e superior da classe neutra definidos para o modelo analisado.

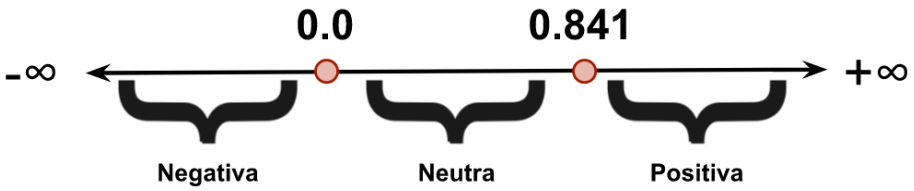

Figura 6.4: *Valores limite das classes para o modelo m_{best}*

Como é possível observar, em m_{best} o limite inferior da classe neutra foi definido pela PG como sendo zero absoluto, ou seja, manteve-se o valor padrão. O limite superior, entretanto, foi alterado, e teve como média 0.841. Isso reforça a adequação feita pelos modelos com relação às fronteiras das classes. Com isso, o resultado $r \in \mathbb{R}$ da avaliação das mensagens $t \in T$ pelo modelo m_{best} segue a regra apresentada a seguir:

$$r = \begin{cases} positivo, & m_{best}(t) > 0.841 \\ neutro, & 0 \leq m_{best}(t) \leq 0.841 \\ negativo, & m_{best}(t) < 0 \end{cases}$$

A análise das saídas do classificador m_{best} pode demonstrar o comportamento geral da solução e como o modelo está atribuindo as classes para as entradas. Como é demonstrado na Figura 6.5, mais de 75% das saídas do modelo são maiores que zero.

Isso não significa, entretanto, que as clases resultantes são todas positivas, uma vez que, como demonstrado nas Figuras 6.3 e 6.4, os limites de valores para a classificação foram alterados.

Esse comportamento reflete diretamente o desbalanceamento da classe negativa nas bases de treinamento e teste, que contém somente 15% e 17% de mensagens desse tipo, respectivamente. Levando essa diferença em consideração, é possível perceber que as adequações realizadas pela PG tendem a priorizar saídas positivas e uma redefinição do limite superior da classe neutra.

É possível perceber que a Programação Genética conseguiu adequar-se ao problema proposto, realizando a ponderação dos dicionários e os ajustes nos limites de cada classe. A técnica mostrou-se competitiva e muitas vezes superior em comparação com outras abordagens clássicas, como demonstrado nos resultados anteriores.

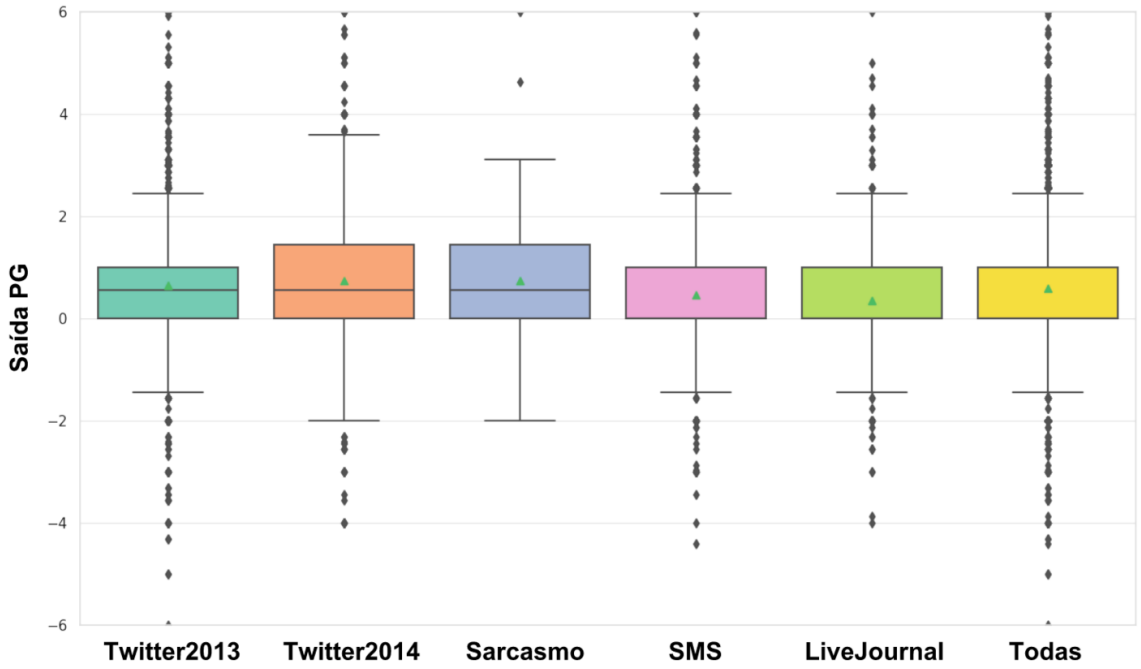

Figura 6.5: *Valores de saída do modelo m_{best} para cada uma das bases de teste*

A PG, entretanto, demanda o maior tempo de treinamento quando comparada com as outras técnicas utilizadas. Isso acontece principalmente pelo fato da necessidade de avaliação de cada um dos indivíduos para o cálculo do respectivo *fitness*. Há algumas formas de tentar minimizar esse impacto, como a paralelização – por sua característica de funcionamento, a PG possui alto grau de paralelismo [56]. Além disso, há variantes da técnica que podem diminuir o custo desses processos, como a *Root Genetic Programming* (RGP), publicada em [66].

6.3 Combinação de classificadores

Como discutido no Capítulo 2, a combinação de diferentes classificadores frequentemente melhora a capacidade de avaliação obtida individualmente pelos modelos. De posse dessa informação, é razoável supor que a associação das técnicas geradas até o momento neste trabalho em um *ensemble* pode incrementar a capacidade de predição da solução, resultando em melhores valores para as bases de teste, superiores aos apresentados na Tabela 6.8.

Como demonstrado, há alguns métodos para estimular a diversidade em comitês de classificadores e um deles tem relação com a utilização de diferentes técnicas de AM [28]. Uma vez que essa abordagem foi a utilizada neste trabalho para a comparação dos resultados da PG, pode-se considerar que os requisitos de pluralidade são atendidos, ou seja, a criação de um *ensemble* com as técnicas desenvolvidas neste trabalho possui a característica de diversidade.

Os modelos gerados pelas técnicas tradicionais de AM utilizadas nesta pesquisa – *Support Vector Machines* (SVM), *Naïve Bayes*, *Random Forest*, Regressão Logística, *Stochastic Gradient Descent* (SGD) – retornam a probabilidade de uma dada entrada pertencer a cada uma das classes disponíveis: positiva, negativa ou neutra. Essas probabilidades serão combinadas na solução de *ensemble*, na busca de obter resultados superiores.

Vale ressaltar que a Programação Genética, principal técnica utilizada neste trabalho, da forma como foi construída, retorna somente o rótulo de predição, ou seja, não dá o resultado em termos de probabilidade de cada classe. Mesmo assim, a PG será incluída no comitê de classificadores, uma vez que obteve bons resultados, sendo superior a todas as outras técnicas em algumas bases de teste, conforme demonstrado na Tabela 6.8. Por isso, o resultado da PG é normalizado em termos de probabilidade, atribuindo o valor 1 para a classe resultante e 0 para as demais.

Tendo como base as técnicas de associação de classificadores propostas em [91], as estratégias de combinação dos modelos utilizados neste trabalho foram: (i) a votação majoritária e (ii) a soma das probabilidades. Na votação majoritária, a classe dominante (maior probabilidade entre as classes disponíveis) retornada pela maioria dos modelos presentes no *ensemble* é escolhida. Como é possível perceber, nesse caso considera-se apenas a predição de cada modelo, ignorando as probabilidades individuais das classes [91].

A soma das probabilidades, como o próprio nome indica, realiza a adição dos valores de predição de cada classe dos n classificadores $c \in C$ conforme apresentado em 6-2.

$$[\sum_{i=1}^{n} c_i pos,\ \sum_{1}^{n} c_i neg,\ \sum_{1}^{n} c_i neu] \tag{6-2}$$

Cada uma das técnicas descritas acima será aplicada na proposta de *ensemble* e uma visão geral da abordagem utilizada é apresentada na Figura 6.6.

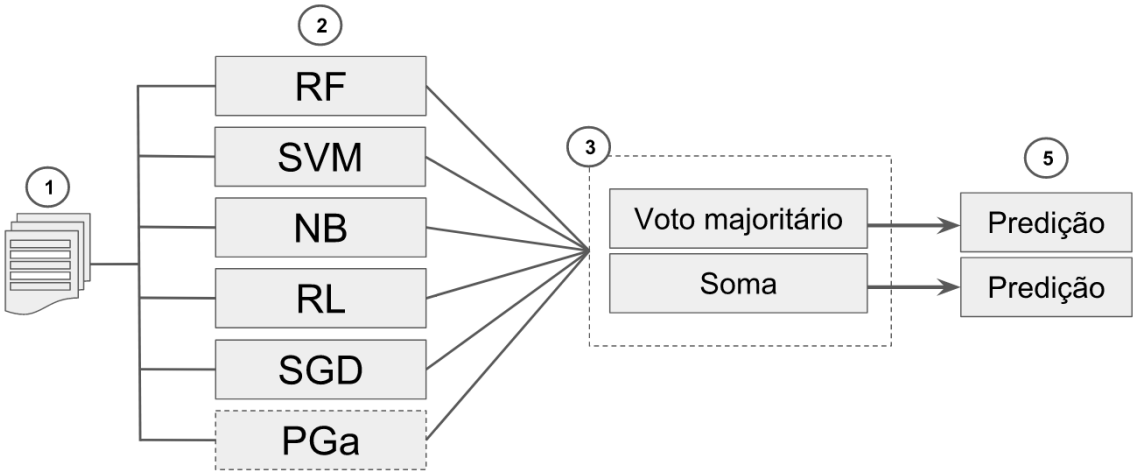

Figura 6.6: *Esquema da proposta de solução usando stacking*

As mensagens da base de teste (item 1) são enviadas como entrada para cada um dos classificadores do *ensemble* (item 2): RF, SVM, NB, RL, SGD e PG (por meio do modelo m_{best} gerado na estratégia PG_a). Os resultados de cada um desses classificadores são combinados por 2 técnicas (item 3): voto majoritário e soma das probabilidades [91]. Cada um desses resultados (item 5) é avaliado por meio das métricas apresentadas no Capítulo 4.

Com o objetivo de identificar a importância geral da PG para o *ensemble*, duas versões do comitê serão implementadas: a primeira delas sem a presença da PG, ou seja, composto somente das outras 5 técnicas de AM utilizadas e a segunda, com a participação da PG_a por meio do melhor modelo m_{best}. Para facilitar a apresentação dos resultados e as referências durante o texto, a solução sem a PG será chamada de $ensemble_{no\_pg}$ e a estratégia posterior, com a inclusão da técnica, referenciada como $ensemble_{pg}$.

Em caso de empate nas estratégias de combinação dos valores de cada classificador de base (item 3), o resultado será definido da seguinte forma: (a) para o $ensemble_{no\_pg}$, os valores resultantes são os indicados por SVM; e (b) para o $ensemble_{pg}$, serão considerados os resultados retornados pela PG_a, por meio do m_{best}. Algumas abordagens para tratamento de empates em comitês de classificadores podem ser vistas em [118, 9, 176].

6.3.1 Comitê de classificadores sem a utilização da PG

Os primeiros resultados obtidos com abordagem utilizando o comitê de classificadores sem a PG ($ensemble_{no\_pg}$) estão compilados na Tabela 6.11. Como é possível observar, a técnica que obteve os melhores resultados foi o voto majoritário e, por isso, serão os valores utilizados para as próximas análises. Em comparação com os resultados

individuais de cada classificador utilizado, é possível perceber que houve acréscimos em praticamente todas as bases de teste, com exceção de Sarcasmo – que havia obtido seus melhores resultados com a técnica de *Naïve Bayes*, com um resultado de 54.25 – e SMS, que atingiu um *F1-score* de 65.25 com SGD.

Tabela 6.11: *Comparação dos resultados (F1-score) das técnicas utilizada no ensemble$_{no_pg}$*

	Técnica	
Base	**Majoritária**	**Soma**
Twitter2013	**66.97**	64.44
Twitter2014	**67.02**	61.08
Sarcasmo	**50.49**	49.64
SMS	**64**	60.4
LiveJournal	**70.21**	66.17
TODAS	**66.99**	63.06

A Tabela 6.12 apresenta os resultados de cada métrica da solução de *ensemble$_{no_pg}$*. É possível identificar que a base de Sarcasmo, assim como em outros testes, obteve os piores resultados, sendo, inclusive, menor que o atingido pela técnica de *Naïve Bayes* individualmente. Isso pode se justificar pelo fato dessa ser a base que resultou nos valores mais baixos em todas as técnicas e a combinação dessas soluções em um comitê faz com que os resultados gerais sejam ainda piores.

Tabela 6.12: *Resultados do ensemble$_{no_pg}$ utilizando a estratégia majoritária*

		Média			
Base	**Acurácia**	**Precisão**	**Revocação**	**F1 P/U/N**	**F1 P/N**
Twitter2013	**71.41**	70.24 ±0.09	70.03 ±0.08	69.5 ±0.05	66.97 ±0.04
Twitter2014	69.45	67.16 ±0.13	69.75 ±0.06	67.31 ±0.05	67.02 ±0.06
Sarcasmo	52.33	56.49 ±0.16	57.5 ±0.21	50.8 ±0.08	50.49 ±0.10
SMS	71.19	68.76 ±0.16	**71.96** ±0.07	68.62 ±0.07	64 ±0.03
LiveJournal	70.23	**71.33** ±0.09	71.43 ±0.09	**70.23** ±0.01	**70.21** ±0.01
TODAS	70.62	69.26 ±0.10	70.65 ±0.06	69.21 ±0.04	66.99 ±0.04

Um aspecto interessante a se notar é o aumento nos valores da métrica de Acurácia em relação aos valores obtidos no F1 P/U/N e F1 P/N. Isso demonstra que o comitê está conseguindo avaliar mais mensagens corretamente, mas tem dificuldades no tratamento do desbalanceamento da base de teste, o que reflete em um *F1-score* menor. Pode-se destacar, por exemplo, a base Twitter2013, que possui o maior valor de Acurácia, com 71.41, ao mesmo tempo que resultou em um *F1-score* de 66.97, considerando as classes positiva e negativa.

A base LiveJournal novamente foi a responsável pelos maiores resultados, seguida da base de Twitter2014 e SMS. Nota-se, assim como ocorrido na base de Sarcasmo, em SMS o valor F1 P/N obtido com o $ensemble_{no\_pg}$ foi menor se comparado com o melhor valor obtido com os classificadores individualmente (diferença de menos de 2% em relação aos valores resultantes por meio da técnica de SGD).

Ao considerar o resultado geral, é possível perceber que o *F1-score* obtido foi superior ao valor resultante de cada técnica individualmente, especificamente 3% maior que o melhor resultado (65.5), obtido por meio da PG_a, como pode ser visto na Tabela 6.8.

6.3.2 Comitê de classificadores com a utilização da PG

A inclusão da PG no comitê de classificadores tem por objetivo a tentativa de incrementar o poder de predição do sistema e, consequentemente, identificar a importância geral da técnica para o *ensemble*. Para a avaliação das mensagens, o melhor modelo gerado pela PG (na versão PG_a), chamado de m_{best}, será considerado. Para facilitar a diferenciação entre as abordagens utilizadas, essa versão do comitê será identificada como $ensemble_{pg}$ nas tabelas e textos que seguem. Os resultados obtidos por cada técnica de combinação são apresentados na Tabela 6.13.

Tabela 6.13: *Comparação dos resultados (F1-score) das técnicas utilizada no ensemble$_{pg}$*

	Técnica	
Base	**Majoritária**	**Soma**
Twitter2013	**68.12**	65.72
Twitter2014	**66.72**	60.99
Sarcasmo	**46.21**	43.95
SMS	**66.24**	61.11
LiveJournal	**74.25**	68.68
TODAS	**68.17**	62.43

Os primeiros números demonstram que, assim como ocorreu em no $ensemble_{no\_pg}$, a técnica de combinação que obteve os melhores resultados foi a majoritária e, por isso, esses valores serão considerados para as demais comparações.

Mais uma vez, os valores obtidos são superiores aos resultados individuais de cada técnica, com exceção da base de Sarcasmo e SMS, essa última com uma diferença de apenas 0.2% com relação ao valor alcançado por SGD. Em LiveJournal, base que já possuía os melhores resultados, houve um incremento de mais de 4% em relação ao melhor valor obtido, conquistado pela PG_a, como demonstrado anteriormente na Tabela 6.8.

Os resultados organizados por métrica, apresentados na Tabela 6.14, demonstram que a base LiveJournal foi a responsável pelos melhores valores de todas as técnicas. Da mesma forma, os piores resultados foram alcançados na base de Sarcasmo sendo, inclusive, inferiores aos obtidos em versões anteriores. A base SMS, mais uma vez, obteve um valor de acurácia elevado, muito próximo do conquistado em LiveJournal. Esse resultado pode ser justificado pela forma como a base é organizada, sendo a que possui a maior disparidade entre as classes, com as mensagens neutras representando 57% do total.

É possível notar, também, que os valores de acurácia aumentaram em comparação com os resultados obtidos em PG_a. Esse comportamento é resultante da combinação dos classificadores, que muitas vezes aumentam a quantidade de mensagens preditas corretamente mas não necessariamente mantém o balanceamento entre as classes, de forma a obter um maior *F1-score*. A PG, por utilizar essa métrica como função objetivo, mostrou-se eficaz na busca por um balanceamento entre as avaliações, muitas vezes permitindo até mesmo uma diminuição no valor de acurácia em detrimento do aumento do *F1-score*.

A queda nos valores conquistados na base de Sarcasmo pode ser resultado da incapacidade da maior parte dos classificadores em lidar com esse tipo de mensagem, refletida diretamente nos resultados individuais para a sua avaliação. Por isso, ao agregar diversos modelos não ajustados para esse tipo de entrada, a técnica que obteve resultados superiores em Sarcasmo (*Naïve Bayes*) não conseguiu impor seus resultados frente aos outros algoritmos. Além disso, é importante perceber que os resultados de Acurácia e F1 P/U/N foram superiores ao F1 P/N, com exceção da base LiveJournal.

Tabela 6.14: *Resultados do $ensemble_{pg}$ utilizando a estratégia majoritária*

		Média			
Base	**Acurácia**	**Precisão**	**Revocação**	**F1 P/U/N**	**F1 P/N**
Twitter2013	71.47	69.91 ±0.07	70.37 ±0.05	69.92 ±0.04	68.12 ±0.04
Twitter2014	69.78	66.66 ±0.12	69.35 ±0.06	67.26 ±0.05	66.72 ±0.07
Sarcasmo	51.16	60.17 ±0.14	58.57 ±0.23	49.33 ±0.10	46.21 ±0.11
SMS	73.01	69.18 ±0.15	73.12 ±0.06	70.43 ±0.06	66.24 ±0.02
LiveJournal	**73.38**	**73.38** ±0.05	**73.78** ±0.03	**73.4** ±0.01	**74.25** ±0.01
TODAS	71.33	69.66 ±0.07	70.66 ±0.03	69.93 ±0.04	68.17 ±0.04

A análise dos resultados das soluções de comitês implementadas demonstra que a abordagem que utiliza a PG como um dos classificadores do conjunto ($ensemble_{pg}$) obteve os melhores resultados em praticamente todas as bases de teste, com exceção de Sarcasmo. Isso reforça a importância do modelo gerado pela PG na predição das classes das mensagens. A maior diferença foi alcançada em LiveJournal, com 5.8% de ganho, seguida de um aumento de 3.5% na avaliação das mensagens pertencentes a base de SMS. Ao considerar a avaliação de todas as mensagens de teste, o aumento foi de 1.8%. A base de Sarcasmo, como comentado anteriormente, atingiu um valor 9.2% inferior se comparado com o comitê sem a PG.

Tabela 6.15: *Comparação dos resultados obtidos por $ensemble_{no\_pg}$ e $ensemble_{pg}$*

	***F1-score* médio classes pos e neg**		
Base	$\boldsymbol{ensemble_{no\_pg}}$	$\boldsymbol{ensemble_{pg}}$	**Diferença**
Twitter2013	66.97	**68.12**	+1.7%
Twitter2014	**67.02**	66.72	-0.4%
Sarcasmo	**50.49**	46.21	-9.2%
SMS	64	**66.24**	+3.5%
LiveJournal	70.21	**74.25**	+5.8%
TODAS	66.99	**68.17**	+1.8%

O teste t pareado demonstra que não se pode rejeitar de pronto a Hipótese nula $H_0 : \mu_1 = \mu_2$, que afirma a média das populações analisadas são iguais. Entretanto, é importante salientar que isso não significa que não há diferenças entre as abordagens, uma vez que elas existem e refletem diretamente nos resultados de predição do sistema, mas tão somente que os resultados do teste são inconclusivos quanto à rejeição de H_0 [49].

A comparação dos melhores resultados alcançados com o comitê de classificadores em relação aos valores individuais obtidos por cada um dos modelos é demonstrada na Tabela 6.16. É possível observar que os resultados conquistados com a utilização do $ensemble_{pg}$ são os melhores para as bases de Twitter2013, Twitter2014, SMS e LiveJournal. Além disso, a solução é superior quando considerada a avaliação de todas as mensagens de teste, obtendo um valor 4% maior que o segundo melhor resultado, encontrado em PG_a.

Esses resultados demonstram que o comitê de classificadores foi capaz de incrementar o poder de predição obtido individualmente pelas técnicas. A combinação das avaliações de cada um dos algoritmos resultou em maiores quantidades de mensagens avaliadas corretamente, além de um incremento geral nas métricas avaliadas. Em especial, o *F1-score* médio das classes positiva e negativa, métrica principal deste trabalho, apresentou ganhos expressivos em todas as bases, com exceção de Sarcasmo.

Tabela 6.16: *Comparação dos resultados (F1-score) entre as técnicas, incluindo os valores de ensemble$_{pg}$*

	F1-score médio classes pos e neg						
Base	***RF***	***SVM***	***NB***	***RL***	***SGD***	***PG$_a$***	***ensemble$_{pg}$***
Twitter2013	49.66	63.75	52.84	61.57	58.5	65.47	**68.12**
Twitter2014	47.5	63.36	51.24	60.48	61.24	62.31	**66.72**
Sarcasmo	43.23	41.38	**54.25**	47.56	35.07	48.04	46.21
SMS	40.89	62.56	43.1	51.32	65.25	62.6	**66.24**
LiveJournal	51.46	69.94	54.61	60.94	68.05	71.24	**74.25**
TODAS	48.73	64.62	51.2	59.32	62.53	65.5	**68.17**

Os testes demonstram que a diferença entre os resultados do *ensemble$_{pg}$* em relação aos melhores valores individuais obtidos (modelo m_{best}, alcançado na PG_a) é estatisticamente significativa, considerando o nível de confiança de 95% adotado no trabalho e o cálculo do teste t pareado. A diferença resulta em um valor de $t = 2.7127$ e, consequentemente, $p = 0.0211$.

A comparação dos resultados obtidos com o comitê *ensemble$_{pg}$* em relação aos trabalhos que fizeram uso do mesmo *benchmark* demonstra que a diferença entre os valores alcançados e as melhores pesquisas ficou ainda menor, com destaque para a base LiveJournal, com valor apenas 0.7% inferior ao primeiro colocado no *ranking* e obtendo o 3º melhor resultado entre os 50 trabalhos submetidos para SemEval 2014.

A maior diferença pode ser observada na base Sarcasmo, com um *F1-score* 25% menor que o melhor resultado do *benchmark* para essa base tendo, inclusive, aumentado a diferença entre os trabalhos com relação à versões anteriores. Como já discutido, esses valores refletem a dificuldade de análise e predição de mensagens que possuem esse tipo de figura de linguagem, apresentando-se como um desafio a ser superado para próximas abordagens. Importante salientar, ainda, que mesmo o melhor trabalho nessa base obteve um *F1-score* de 58.16, valor 20% menor que o maior valor obtido para SMS, por exemplo, base que obteve o segundo menor valor de primeiro colocado.

Em Twitter2013 e Twitter2014, observa-se uma diferença de 5.8% e 6.3% em relação ao primeiro colocado, respectivamente. Cabe citar que, considerando a diferença necessária para alcançar os 3 primeiros classificados no *benchmark*, esses valores caem para 3.3% e 4.8%, respectivamente. Na base de SMS, percebe-se uma diferença de 6% em relação ao melhor resultado e apenas 1% inferior aos 3 primeiros trabalhos.

Tabela 6.17: *Comparação de resultados obtidos em $ensemble_{pg}$ com os trabalhos submetidos para SemEval 2014 (F1-score)*

Base	Resultado $ensemble_{pg}$	Majority Baseline	Top 3 SemEval
Twitter2013	68.12	29.2	1º 72.12 2º 70.75 3º 70.40
Twitter2014	66.72	34.6	1º 70.96 2º 70.14 3º 69.95
Sarcasmo	46.21	27.7	1º 58.16 2º 57.26 3º 56.50
SMS	66.24	19	1º 70.28 2º 67.68 3º 67.51
LiveJournal	74.25	27.2	1º 74.84 2º 74.46 3º 73.99

Os resultados apresentados indicam que a combinação das técnicas utilizadas neste trabalho, incluindo a PG, em uma abordagem de comitê de classificadores foi capaz de incrementar substancialmente o poder de predição do sistema, resultando nos melhores valores pra praticamente todas as bases de teste, com exceção das mensagens de Sarcasmo. Esses resultados indicam que a suposição levantada no início da Seção 6.3, de que a associação das técnicas utilizadas poderia melhorar a capacidade geral da solução, mostrou-se verdadeira.

A Tabela 6.18 compila o *F1-score* das soluções apresentadas nesse Capítulo. É possível observar que a atualização no processo de treinamento da PG, chamada de PG_a, foi capaz de incrementar o poder de predição da solução. Além disso, a combinação das técnicas por meio de um comitê de classificadores foi capaz de aumentar, de forma geral, a eficácia do sistema.

Em sua primeira versão, sem a utilização da PG como um dos classificadores do conjunto (identificada como $ensemble_{no\_pg}$), o sistema foi capaz de aumentar os resultados das predições em praticamente todas as bases, com exceção de LiveJournal,

com um resultado 1.4% menor que o alcançado em PG_a, porém com um resultado geral 2.2% maior em relação ao maior valor obtido anteriormente.

Tabela 6.18: *Comparação dos melhores resultados obtidos com a PG*

	F1-score médio classes pos e neg			
Base	**PG**	PG_a	$ensemble_{no\_pg}$	$ensemble_{pg}$
Twitter2013	62.78	65.47	66.97	**68.12**
Twitter2014	61.31	62.31	67.02	**66.72**
Sarcasmo	48.38	48.04	**50.49**	46.21
SMS	60.38	62.6	64	**66.24**
LiveJournal	68.53	71.24	70.21	**74.25**
TODAS	62.33	65.5	66.99	**68.17**

A atualização do comitê com a inclusão da PG, identificada com $ensemble_{pg}$, foi capaz de aumentar os ganhos obtidos pela versão anterior ($ensemble_{no\_pg}$), inclusive superando os resultados da base LiveJournal alcançados pela PG_a. A exceção mais uma vez foi observada em Sarcasmo, que resultou em valores menores que os encontrados em $ensemble_{no\_pg}$, PG_a e PG. Esse resultado reflete a dificuldade das técnicas em lidar com esse tipo de mensagem, o que demanda um estudo mais detalhado para as próximas abordagens, com o objetivo de aumentar o poder de predição dessas entradas.

CAPÍTULO 7

Considerações finais

Pesquisas na área de Análise de Sentimentos (AS) vêm avançando nos últimos anos motivadas, principalmente, pela popularização da Internet e o consequente aumento do volume de conteúdo gerado na rede. Esse fenômeno aumentou substancialmente a quantidade de dados e informações disponíveis para o treinamento e teste de técnicas de classificação de sentimentos. Ao mesmo tempo, a evolução dos métodos de Aprendizado de Máquina (AM) e o aumento da disponibilidade e barateamento de *hardware* para processamento desses métodos faz com que melhores resultados sejam obtidos ano após ano.

Considerando o momento atual, classificadores de sentimentos possuem um papel fundamental para empresas, governos, pessoas, etc. Saber o sentimento geral sobre determinado assunto, produto ou pessoa, ou até mesmo identificar automaticamente alguns tipos de mensagens mostra-se um ativo importante e apresenta-se como um diferencial em um mercado cada vez mais competitivo e complexo e em um mundo cada dia mais conectado [106].

Sabendo disso, o presente trabalho tem como principal propósito a criação de classificadores de sentimentos eficazes, que maximizem a quantidade de avaliações corretas utilizando para isso uma abordagem híbrida, ou seja, por meio da utilização de dicionários léxicos – conjunto de palavras e suas respectivas polaridades – e de técnicas de Aprendizado de Máquina, inclusive combinadas para aumentar a eficácia de classificação.

A técnica principal escolhida para a pesquisa foi a Programação Genética (PG), um algoritmo pertencente ao conjunto das abordagens evolucionárias, que se baseiam nas teorias da evolução de *Charles Darwin*, com uma população de indivíduos que evolui a cada geração, priorizando os de melhor aptidão para determinado domínio [93, 145].

A PG vem obtendo bons resultados na resolução de problemas do mundo real, muitas vezes superando soluções encontradas por especialistas humanos [145, 66]. A escolha da técnica deu-se, principalmente, pela simplicidade e facilidade de utilização, demandando pouco conhecimento sobre a resolução do problema. Além disso, a forma como apresenta seus modelos – por meio de árvores, na maioria dos casos – é muito interessante, pois possibilita a validação de conhecimento sobre o domínio de análise,

bem como o aprendizado, pois permite a leitura dos passos para a solução.

O Capítulo 3 demonstra que poucas pesquisas utilizam estratégias evolucionárias para a resolução de problemas na área de AS. Apesar disso, acredita-se que essas técnicas podem acarretar bons resultados, compatíveis ou superiores às abordagens clássicas utilizadas.

No Capítulo 4, a criação automatizada de um classificador de sentimentos foi formalizada como um problema de otimização, com o objetivo de encontrar um bom modelo de classificação em meio a um conjunto de possíveis modelos de forma a maximizar o poder de predição das entradas. Foram definidos, em consonância com o contexto do problema, os principais itens da PG: o conjunto de primitivas P, composta de funções $f \in F$ e terminais $t \in T$, função de aptidão, parâmetros de controle e os critérios de parada.

De posse da ideia principal de solução para o problema proposto, duas frentes principais de trabalho foram definidas. A primeira delas trata da criação de uma solução híbrida de classificação de sentimentos por meio da utilização da Programação Genética, e a segunda aborda a combinação de diferentes técnicas de AM, juntamente com a PG, em busca de um incremento na eficácia da classificação. Considerando a primeira frente do trabalho, as Hipóteses de Pesquisa levantadas foram:

- Classificadores híbridos de sentimentos inferidos automaticamente com o uso de Programação Genética apresentam resultados competitivos ou superiores aos valores obtidos a partir de técnicas clássicas de Aprendizado de Máquina.
- Classificadores híbridos de sentimentos inferidos por meio de Programação Genética podem ser usados para, a partir de vários léxicos disponíveis na literatura, aprender e escolher quais são os mais relevantes para o domínio em questão.

O Capítulo 5 descreve o *benchmark* SemEval 2014, utilizado para a validação das hipóteses deste trabalho, que fornece uma base de treinamento, composta de 9.684 mensagens, e um conjunto de teste, com 8.987 entradas, classificadas em positivo, negativo e neutro [163]. Além disso, é disponibilizado o resultado dos trabalhos submetidos à competição, característica relevante principalmente para a primeira Hipótese de Pesquisa, uma vez que é essencial a comparação com outras técnicas para identificar se os resultados são competitivos ou não.

Dicionários foram selecionados para serem utilizados na pesquisa, escolhidos com base nos trabalhos relacionados (detalhados no Capítulo 3, Tabela 3.2). Esses léxicos mostraram-se atributos importantes na maior parte das pesquisas que alcançaram os melhores resultados no *benchmark* e também são imprescindíveis para a validação da segunda Hipótese do trabalho. Ao todo, 11 dicionários foram selecionados para este estudo, e são descritos em detalhes na Seção 4.5.

Devido a natureza estocástica da PG, 30 modelos foram gerados para cada cenário estabelecido. Essa estratégia tem por objetivo principal garantir que os modelos não foram gerados puramente por processos aleatórios e que os resultados convergem para um limiar comum. Para medir a qualidade de cada um dos modelos gerados, as principais métricas de avaliação de classificadores foram utilizadas, além da medição da média e do desvio padrão entre eles [158]. Além disso, as diferenças entre os resultados obtidos durante o trabalho foram avaliadas por meio do teste t pareado, com um intervalo de confiança de 95% [78]. O Capítulo 4 apresenta cada uma das medidas em detalhes, bem como as respectivas justificativas para a sua utilização.

Alguns parâmetros utilizados na configuração da PG deste trabalho não são tão usuais na literatura, como a quantidade de gerações maior que 51. A alteração na medida foi feita após testes empíricos e análise do comportamento dos indivíduos, com a hipótese de que ainda havia espaço para evoluções no modelo sem que ocorresse o sobreajuste. De todo modo, é possível encontrar na literatura diversos trabalhos que utilizam uma parametrização com número de gerações maior que o de indivíduos para a resolução de problemas por meio da PG [107, 186, 145]. Os principais parâmetros utilizados neste trabalho foram demonstrados no Capítulo 4, Tabela 4.3.

Os primeiros resultados indicaram que o classificador de sentimentos criado com a PG possui resultados competitivos quando comparado com os trabalhos que fizeram uso do mesmo *benchmark*. Com destaque à base LiveJournal, que obteve os melhores valores, resultando em um F1 P/N – média do *F1-score* considerando as classes positiva e negativa – de 68.53, apenas 6.31 pontos do primeiro colocado e 41.33 pontos acima do *Majority Baseline*.

Os piores resultados foram obtidos na base Sarcasmo, com F1 P/N de apenas 48.38. Apesar disso, é possível observar nos trabalhos relacionados e nos modelos gerados com as técnicas clássicas de AM, que os resultados gerais para essa base são muito baixos. Além disso, na pesquisa publicada em [63], o autor demonstra que a habilidade de detecção de sarcasmo por humanos obteve uma acurácia de apenas 62.59%. Tudo isso confirma a dificuldade em lidar com mensagens que possuem esse tipo de figura de linguagem, sendo considerada um dos grandes desafios da AS [149, 106].

Quando comparada com algoritmos clássicos de Aprendizado de Máquina – *Support Vector Machine* (SVM), *Naïve Bayes* (NB), *Random Forest* (RF), Regressão Logística (RL) e *Stochastic Gradient Descent* (SGD) – a solução fornecida pela PG é superior à 2 de 5 técnicas em todas as bases, o que demonstra que a Programação Genética, apesar de pouco usada no contexto de AS, pode entregar resultados satisfatórios e competitivos. O SVM alcançou os melhores resultados, com um F1 P/N de 64.62 na avaliação de todas as mensagens, pouco mais de 3% superior ao atingido pela PG, de 62.33. Os detalhes da comparação entre as técnicas são compilados na Tabela 6.4 do

Capítulo 6.

Em comparação com SVM, observou-se que a PG obteve resultados inferiores em 4 de 5 subconjuntos de testes. Apesar disso, essa diferença é pequena, sendo a maior delas 2.18 pontos, e há casos em que a utilização da PG se justifica mesmo com a diferença de resultados, como em situações em que se pretende conhecer como o modelo atribui polaridade às mensagens e quando há interesse em um processo automatizado de seleção de *features*, algo inerente do funcionamento da PG [66].

A generalização, uma das principais características de um bom classificador, foi motivo de preocupação durante a pesquisa. Pelo fato da PG utilizar a base de treinamento em sua totalidade para a criação da solução, foi levantada a hipótese de que os modelos pudessem estar se especializando em demasia na classificação das mensagens de treinamento, fenômeno conhecido como *overfitting*.

Para avaliar essa hipótese, alterações foram feitas no fluxo geral da PG (ilustradas na Figura 6.1 do Capítulo 6), e uma nova versão foi criada – chamada no trabalho de PG_a. A principal mudança ocorreu na forma de treinamento dos modelos, com a divisão do conjunto de dados em partes iguais e a consequentemente alteração da função de aptidão para considerar a média do *F1-score* das classes positiva e negativa, como formalizado na Equação 6-1.

Os testes da PG_a demonstraram um crescimento nos resultados de praticamente todas as bases – com exceção de Sarcasmo, que obteve um decréscimo de 0.7% – o que sugere que os modelos gerados conseguiram generalizar o processo de predição de forma mais adequada. Os resultados obtidos mostram que a PG_a é superior às técnicas de RF e RL em todas as bases, além de possuir os melhores resultados entre todos os algoritmos nos conjuntos de Twitter2013 e LiveJournal.

Além disso, a PG_a alcança os melhores valores na avaliação de todas as mensagens de teste, com um F1 P/N de 65.5, aproximadamente 2% maior que o obtido pelo segundo melhor algoritmo (SVM). Esses resultados reforçam a competitividade da abordagem utilizada, evidenciando a validade da primeira Hipótese de Pesquisa levantada. Os detalhes das avaliações de cada técnica em conjunto com a PG_a são apresentados na Tabela 6.7 do Capítulo 6.

A combinação dos dicionários é uma característica importante do trabalho, e os resultados demonstraram que a PG foi capaz de atribuir valores de importância para cada um dos 11 léxicos utilizados na solução. Além de indicar a validade da segunda Hipótese de Pesquisa levantada, essa funcionalidade mostra-se fundamental para entender a relevância de determinado conjunto de palavras no contexto em que estão sendo utilizadas. Os pesos médios atribuídos para cada um dos dicionários utilizados foi demonstrado na Figura 6.2.

É possível visualizar que os léxicos Effect (w_6) e NRC (w_8) obtiveram os

menores pesos, com valores abaixo de 0.001, para todos os modelos gerados pela PG. É importante salientar que a baixa relevância de um dicionário em determinado domínio não implica que o mesmo atribui polaridades incorretas às palavras, mas tão somente que o léxico não foi expressivo para determinado *benchmark*. O dicionário AFINN (w_3) obteve os maiores valores de peso, ou seja, foi capaz de auxiliar de forma mais eficaz na maximização das predições dos modelos, seguido de VADER (w_4) e LIU (w_1).

A definição dos valores das fronteiras da classe neutra, uma das responsabilidades da PG, também foi uma característica analisada, de forma a entender o processo de atribuição de polaridade pelos modelos. A configuração desses valores, feita por meio da função *neutralRange*, define os limites inferior (*IR*) e superior (*SR*) da classe neutra, como ilustrado na Figura 4.4.

Foi possível perceber que houve uma tendência dos modelos gerados em elevar os limites inferior e superior, sendo 75% dos valores definidos para *IR* maiores que zero e, para *SR*, 100% dos casos. Isso demonstrou, também, que a PG julgou mais eficaz uma alteração nesses limites em detrimento da utilização da avaliação padrão, que definia somente o valor zero para a classe neutra (demonstrada na Seção 4.7).

A matriz de confusão dos resultados obtidos fornece boas pistas para a identificação de pontos críticos e que podem ser explorados para obter melhores resultados. O principal problema identificado foi a quantidade de mensagens neutras sendo classificadas como positiva, como pode ser visto na Tabela 6.5, Capítulo 6. Isso demonstra que, em algumas oportunidades, o classificador está atribuindo polaridade em demasia para as mensagens. Também pode refletir um ajuste insuficiente nos limites de cada uma das classes.

É importante salientar que a simples alteração nos limiares das classes pode acarretar em uma piora no resultado geral do classificador, uma vez que, mesmo que o problema anterior seja corrijido, não há garantias de que as mensagens que haviam sido corretamente classificadas continuarão com esse *status*. Além disso, como a métrica principal do trabalho é o *F1-score* médio das classes positiva e negativa, não basta somente avaliar corretamente novas mensagens (refletindo na acurácia) mas, sim, deve haver um balanceamento entre as classes disponíveis e avaliações realizadas.

Uma vez que diferentes técnicas de Aprendizado de Máquina foram utilizadas no trabalho para a comparação dos resultados obtidos pela Programação Genética, a segunda frente de trabalho tem por objetivo a combinação dessas abordagens em um comitê de classificadores, com o propósito de incrementar o poder de predição do sistema, resultando em valores superiores aos alcançados individualmente por cada técnica. Decorrente disso, uma nova Hipótese de Pesquisa foi formulada:

- A utilização da PG, em conjunto com outras técnicas de Aprendizado de Máquina, organizados em um comitê de classificadores pode incrementar o poder de predição

e o resultado geral do sistema de AS, resultando em valores superiores aos alcançados considerando os resultados individuais da PG e dos outros métodos.

Em busca da validação da Hipótese supracitada, duas versões de comitês foram construídas: uma delas utilizando todas as técnicas de AM empregadas no trabalho, com exceção da PG, e outra com a inclusão do melhor modelo gerado com a Programação Genética, chamado de m_{best}, no conjunto de classificadores. As técnicas utilizadas para a combinação dos resultados de cada modelo foram a votação majoritária e soma das probabilidades [91], com a primeira obtendo os melhores resultados. Os valores alcançados em cada uma das combinações foram apresentados nas Tabelas 6.11 e 6.13.

Foi possível observar um aumento generalizado nos resultados de todas as bases de teste com a combinação dos classificadores. A primeira versão, chamada de $ensemble_{no\_pg}$, foi superada pela variante que inclui o modelo m_{best} gerado pela PG, chamada de $ensemble_{pg}$, em praticamente todas as bases, com exceção de Sarcasmo e Twitter2014, como pode ser visto na Tabela 6.15.

Em comparação com os trabalhos do *benchmark*, é interessante destacar os resultados alcançados na base LiveJournal, com valor apenas 0.7% inferior ao primeiro colocado no *ranking* e obtendo a 3º melhor marca entre os 50 trabalhos submetidos para SemEval 2014. Os resultados obtidos sugerem que a terceira Hipótese de Pesquisa levantada é válida, uma vez que os valores são superiores aos conquistados individualmente por cada uma das técnicas. A comparação dos resultados alcançados pode ser vista na Tabela 6.16.

Como se pode observar, o objetivo de geração de um classificador híbrido para Análise de Sentimentos utilizando a Programação Genética e a combinação de dicionários léxicos foi atingido. Além disso, ao analisar os resultados obtidos pela técnica, pode-se considerar que a primeira Hipótese de Pesquisa se confirma, ou seja, os classificadores criados com a PG são competitivos com as saídas geradas por técnicas clássicas de AM.

Foi possível observar, também, que a PG permitiu a definição da relevância dos dicionários – por meio de seus pesos – para o domínio do problema, validando a segunda Hipótese de Pesquisa levantada no trabalho. Por fim, a combinação da PG com outras técnicas de AM foi capaz de incrementar o poder de predição do sistema, resultando em um classificador híbrido mais eficaz, o que valida a terceira Hipótese de Pesquisa levantada.

Dentre as principais contribuições desta pesquisa, é possível citar a criação de um método de geração automatizada de modelos híbridos de classificação de sentimentos por meio da Programação Genética e dicionários, capaz de atribuir pesos aos léxicos utilizados, representando sua relevância para o domínio em que está sendo utilizado. Além disso, a solução permite a agregação de outras técnicas de AM, com o objetivo de aumentar a eficácia da classificação das mensagens.

Os resultados obtidos pela solução e por cada uma das técnicas isoladamente também podem ser considerados contribuições relevantes, uma vez que abrange diversos algoritmos de Aprendizado de Máquina e os processos para a agregação das soluções em um esquema de comitê de classificadores.

A compilação dos trabalhos relacionados da área de Análise de Sentimentos, suas técnicas, atributos e dicionários léxicos utilizados (demonstrados nas Tabelas 3.1, 3.2 e 3.3 do Capítulo 3) pode ser considerada uma contribuição relevante, uma vez que fornece uma visão geral das pesquisas desenvolvidas na área, ao mesmo tempo que facilita a análise das principais características em comum entre eles.

O impacto da parametrização na Programação Genética é um assunto importante, alvo de diversas discussões e estudos na literatura [40, 93, 145]. Os testes empíricos realizados neste trabalho contribuíram para a avaliação de combinações paramétricas na PG e suas consequências. As implementações iniciais desta pesquisa resultaram, ainda, em um artigo [24], publicado e apresentado no XIII Congresso Brasileiro de Inteligência Computacional (CBIC 2017)[1].

Os códigos desenvolvidos, os modelos gerados e os arquivos de configuração utilizados para o desenvolvimento desta dissertação estão disponíveis sob a licença GPL[2] no github[3]. Os *datasets* e alguns dicionários não são disponibilizados no repositório devido a termos de uso, mas podem ser encontrados para *download* nos sites oficiais.

7.1 Melhorias futuras

A inclusão de novos dicionários pode melhorar ainda mais os resultados obtidos pela PG, uma vez que se caracterizam como recursos essenciais para o apoio na descoberta de polaridades das mensagens. Além disso, uma vez que a técnica apresenta resultados eficazes na detecção da importância dos léxicos (por meio da definição de pesos), a adição de um possível dicionário não adequado ao contexto não causa impactos significativos no resultado final, uma vez que a técnica é capaz de não considerá-lo na análise, assim como fez com alguns léxicos, conforme demonstrado na Figura 6.2.

A estratégia de *ensemble* empregada neste trabalho faz uso da combinação dos classificadores por meio das técnicas de voto majoritário e soma das probabilidades. Apesar de ter obtidos bons resultados – maiores que os atingidos por cada classificador individualmente – é possível que alguns arranjos mais sofisticados obtenham resultados ainda melhores. Isso pode ser feito, por exemplo, utilizando um esquema de *stacking*, como

[1] http://cbic2017.org/papers/cbic-paper-13.pdf
[2] https://opensource.org/licenses/GPL-3.0
[3] https://github.com/airtonbjunior/opinionMining

apresentado no Capítulo 2, posicionando os modelos gerados no nível 0 – como classificadores de base – e aplicando a própria Programação Genética como meta classificador. A PG, por sua forma de trabalho, pode implementar arranjos otimizados das funções de combinação dos classificadores, além da atribuição de peso em cada um deles, representando sua importância para o comitê.

O custo de treinamento da PG é o maior entre todos os algoritmos utilizados neste trabalho. Isso decorre, principalmente, da necessidade da execução de todos os indivíduos de forma a obter os valores de *fitness*. Tentativas de melhoria no tempo de treinamento dos modelos podem passar pela utilização de variações da PG original, como a *Root Genetic Programming* (RGP), publicada em [66], em que todos os operadores genéticos são aplicados ao nó raíz e somente a ele. Ainda, por conta da característica altamente paralela dos processos de evolução, a PG pode beneficiar-se fortemente com a paralelização do processamento, e há diversas pesquisas que tratam sobre o assunto, como [146, 135, 8, 56].

Referências Bibliográficas

[1] ABBASI, A.; CHEN, H.; SALEM, A. **Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums**. *ACM Transactions on Information Systems (TOIS)*, 26(3):12, 2008.

[2] AGARWAL, A.; XIE, B.; VOVSHA, I.; RAMBOW, O.; PASSONNEAU, R. **Sentiment analysis of twitter data**. In: *Proceedings of the workshop on languages in social media*, p. 30–38. Association for Computational Linguistics, 2011.

[3] AGARWAL, B.; MITTAL, N.; BANSAL, P.; GARG, S. **Sentiment analysis using common-sense and context information**. *Computational intelligence and neuroscience*, 2015:30, 2015.

[4] AHMAD, M.; AFTAB, S.; ALI, I.; HAMEED, N. **Hybrid tools and techniques for sentiment analysis: A review**. In: *INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY SCIENCES AND ENGINEERING*, volume 8, 2017.

[5] AISOPOS, F.; PAPADAKIS, G.; VARVARIGOU, T. **Sentiment analysis of social media content using n-gram graphs**. In: *Proceedings of the 3rd ACM SIGMM international workshop on Social media*, p. 9–14. ACM, 2011.

[6] AL-MANNAI, K.; ALSHIKHABOBAKR, H.; WASI, S. B.; NEYAZ, R.; BOUAMOR, H.; MOHIT, B. **Cmuq-hybrid: Sentiment classification by feature engineering and parameter tuning**. In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, p. 181–185, 2014.

[7] ALASMARI, S. F.; DAHAB, M. **Sentiment detection, recognition and aspect identification**. *International Journal of Computer Applications*, p. 31–37, 2017.

[8] ALBA, E.; TOMASSINI, M. **Parallelism and evolutionary algorithms**. *IEEE transactions on evolutionary computation*, 6(5):443–462, 2002.

[9] ALBUQUERQUE, R. A. S.; OTHERS. **Seleção dinâmica de comitês de classificadores baseada em diversidade e acurácia para detecção de mudança de conceitos**. 2018.

[10] ALENCAR, R. **Estudo da ocorrência de cyberbullying contra professores na rede social twitter por meio de um algoritmo de classificação bayesiano**. In: *Texto Livre: Linguagem e Tecnologia*, volume 5, 2012.

[11] ALMEIDA, A. W. **Sentiment Analysis in Short Messages Using Affective Lexicons**. Master's thesis, Pontifícia Universidade Católica do Paraná - PUC/PR, Curitiba, PR, Brasil.

[12] ALMEIDA, A. W. **Sentiment analysis in short messages using affective lexicons**, 2017.

[13] ARAQUE, O.; CORCUERA-PLATAS, I.; SANCHEZ-RADA, J. F.; IGLESIAS, C. A. **Enhancing deep learning sentiment analysis with ensemble techniques in social applications**. *Expert Systems with Applications*, 77:236–246, 2017.

[14] ARAUJO, G. D. **Análise de sentimento de mensagens do twitter em português brasileiro relacionadas a temas de saúde**. 2014.

[15] ARAÚJO, M.; GONÇALVES, P.; BENEVENUTO, F. **Métodos para análise de sentimentos no twitter**. 2013.

[16] ARORA, S.; MAYFIELD, E.; PENSTEIN-ROSÉ, C.; NYBERG, E. **Sentiment classification using automatically extracted subgraph features**. In: *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, p. 131–139. Association for Computational Linguistics, 2010.

[17] BALAGE FILHO, P. P.; AVANÇO, L. V.; PARDO, T. A. S.; NUNES, M. D. G. V.; OTHERS. **Nilc_usp: an improved hybrid system for sentiment analysis in twitter messages.** In: *International Workshop on Semantic Evaluation, 8th*. ACL Special Interest Group on the Lexicon-SIGLEX, 2014.

[18] BARBOSA, L.; FENG, J. **Robust sentiment detection on twitter from biased and noisy data**. In: *Proceedings of the 23rd international conference on computational linguistics: posters*, p. 36–44. Association for Computational Linguistics, 2010.

[19] BASARI, A. S. H.; HUSSIN, B.; ANANTA, I. G. P.; ZENIARJA, J. **Opinion mining of movie review using hybrid method of support vector machine and particle swarm optimization**. *Procedia Engineering*, 53:453–462, 2013.

[20] BECKER, K.; TUMITAN, D. **Introdução à mineração de opiniões: Conceitos, aplicações e desafios**. *Simpósio brasileiro de banco de dados*, 2013.

[21] BECKER, L.; ERHART, G.; SKIBA, D.; MATULA, V. **Avaya: Sentiment analysis on twitter with self-training and polarity lexicon expansion**. In: *Second Joint Conference on Lexical and Computational Semantics (* SEM)*, volume 2, p. 333–340, 2013.

[22] BENEVENUTO, F.; RIBEIRO, F.; ARAÚJO, M. **Métodos para análise de sentimentos em mídias sociais**, 2015.

[23] BOLLEN, J.; MAO, H.; ZENG, X. **Twitter mood predicts the stock market**. *Journal of computational science*, 2(1):1–8, 2011.

[24] BORDIN-JR, A.; CAMILO-JR, C.; FELIX, N.; ROSA, T. **Aplicando programação genética na geração de classificadores de sentimento**. *Congresso Brasileiro de Inteligência Computacional (CBIC 2017)*, 2017.

[25] BOTTOU, L. **Stochastic gradient descent tricks**. In: *Neural networks: Tricks of the trade*, p. 421–436. Springer, 2012.

[26] BREIMAN, L. **Bagging predictors**. *Machine learning*, 24(2):123–140, 1996.

[27] BREIMAN, L. **Random forests**. *Machine learning*, 45(1):5–32, 2001.

[28] BROWN, G.; WYATT, J.; HARRIS, R.; YAO, X. **Diversity creation methods: a survey and categorisation**. *Information Fusion*, 6(1):5–20, 2005.

[29] BUGEJA, R. **Twitter Sentiment Analysis for Marketing Research**. PhD thesis, University of Malta, 2014.

[30] CAMPOS, C. M.; OTHERS. **Comitê de classificadores em bases de dados transacionais desbalanceadas com seleção de características baseada em padrões minerados**. 2016.

[31] CHEHOURI, A.; YOUNES, R.; PERRON, J.; ILINCA, A. **A constraint-handling technique for genetic algorithms using a violation factor**. *arXiv preprint arXiv:1610.00976*, 2016.

[32] CHEN, L.; WANG, W.; SHETH, A. P. **Are twitter users equal in predicting elections? a study of user groups in predicting 2012 us republican presidential primaries**. In: *International Conference on Social Informatics*, p. 379–392. Springer, 2012.

[33] CHOI, Y.; WIEBE, J. **+/-effectwordnet: Sense-level lexicon acquisition for opinion inference**. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 1181–1191, 2014.

[34] CLARK, S.; WICENTWOSKI, R. **Swatcs: Combining simple classifiers with estimated accuracy**. In: *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, volume 2, p. 425–429, 2013.

[35] CORTES, C.; VAPNIK, V. **Support-vector networks**. *Machine learning*, 20(3):273–297, 1995.

[36] COUTINHO, E.; MOREIRA, L.; PAILLARD, G.; DE LIMA, E. T. **Análise do sentimento de mensagens de chats em uma turma de graduação de um curso de educação à distância**. In: *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, volume 5, p. 1019, 2016.

[37] DA SILVA, N. F.; HRUSCHKA, E. R.; HRUSCHKA JR, E. R. **Tweet sentiment analysis with classifier ensembles**. *Decision Support Systems*, 66:170–179, 2014.

[38] DA SILVA, W. K. N.; DE MEDEIROS SANTOS, A. **Estratégias de construções de comitês de classificadores multirrótulos no aprendizado semissupervisionado multidescrição**. *Revista de Informática Teórica e Aplicada*, 24(2):71–100, 2017.

[39] DE AGUIAR, E. J.; FAIÇAL, B. S.; UEYAMA, J.; SILVA, G. C.; MENOLLI, A. **Análise de sentimento em redes sociais para a língua portuguesa utilizando algoritmos de classificação**. In: *Simpósio Brasileiro de Redes de Computadores (SBRC)*, volume 36, 2018.

[40] DE CARVALHO, M. G.; LAENDER, A. H.; GONÇALVES, M. A.; PORTO, T. C. **The impact of parameter setup on a genetic programming approach to record deduplication**. In: *Proceedings of the 23rd Brazilian symposium on Databases*, p. 91–105. Sociedade Brasileira de Computação, 2008.

[41] DE OLIVEIRA BATISTA, J.; RODRIGUES, R. B.; VAREJAO, F. M. **Um comitê de classificadores svm para diagnóstico de falhas em motobombas baseado em técnicas de soft-computing**. *Simpósio Brasileiro de Pesquisa Operacional*, 2016.

[42] DE SOUZA, K. F.; PEREIRA, M. H. R.; DALIP, D. H. **Unilex: Método léxico para análise de sentimentos textuais sobre conteúdo de tweets em português brasileiro**. *Abakós*, 5(2):79–96, 2017.

[43] DEHZANGI, A.; KARAMIZADEH, S. **Solving protein fold prediction problem using fusion of heterogeneous classifiers**. *INFORMATION, An International Interdisciplinary Journal*, 14(11):3611–3622, 2011.

[44] DIAKOPOULOS, N. A.; SHAMMA, D. A. **Characterizing debate performance via aggregated twitter sentiment**. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, p. 1195–1198. ACM, 2010.

[45] DIETTERICH, T. G. **Ensemble methods in machine learning**. In: *International workshop on multiple classifier systems*, p. 1–15. Springer, 2000.

[46] DING, X.; LIU, B.; YU, P. S. **A holistic lexicon-based approach to opinion mining**. In: *Proceedings of the 2008 international conference on web search and data mining*, p. 231–240. ACM, 2008.

[47] DUWAIRI, R.; AHMED, N. A.; AL-RIFAI, S. Y. **Detecting sentiment embedded in arabic social media–a lexicon-based approach**. *Journal of Intelligent & Fuzzy Systems*, 29(1):107–117, 2015.

[48] ESCALANTE, H. J.; GARCÍA-LIMÓN, M. A.; MORALES-REYES, A.; GRAFF, M.; MONTES-Y GÓMEZ, M.; MORALES, E. F.; MARTÍNEZ-CARRANZA, J. **Term-weighting learning via genetic programming for text classification**. *Knowledge-Based Systems*, 83:176–189, 2015.

[49] ESPÍRITO SANTO, H.; DANIEL, F. **Calcular e apresentar tamanhos do efeito em trabalhos científicos (1): As limitações do p< 0, 05 na análise de diferenças de médias de dois grupos (calculating and reporting effect sizes on scientific papers (1): P< 0.05 limitations in the analysis of mean differences of two groups)**. 2017.

[50] ESULI, A.; SEBASTIANI, F. **Sentiwordnet: a high-coverage lexical resource for opinion mining**. *Evaluation*, 2007.

[51] EVERT, S.; PROISL, T.; GREINER, P.; KABASHI, B. **Sentiklue: Updating a polarity classifier in 48 hours**. In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, p. 551–555, 2014.

[52] FELDMAN, R. **Techniques and applications for sentiment analysis**. *Commun. ACM*, 56(4):82–89, Apr. 2013.

[53] FILHO, J. A. C. **MineraÇÃo de textos: Análise de sentimento utilizando tweets referentes À copa do mundo 2014**, 2015.

[54] FLAVIO, C.; ALVAREZ, G. M.; GONÇALVES, A. L. **Análise de sentimento e mineração de opinião: uma revisão bibliométrica da literatura**. *Análise*, 38(14), 2017.

[55] FLEKOVA, L.; FERSCHKE, O.; GUREVYCH, I. **Ukpdipf: Lexical semantic approach to sentiment polarity prediction in twitter data**. In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, p. 704–710, 2014.

[56] FOGEL, D. B. **The advantages of evolutionary computation.** 1997.

[57] FORTIN, F.-A.; DE RAINVILLE, F.-M.; GARDNER, M.-A.; PARIZEAU, M.; GAGNÉ, C. **DEAP: Evolutionary algorithms made easy**. *Journal of Machine Learning Research*, 13:2171–2175, jul 2012.

[58] FREITAS, C. **Sobre a construção de um léxico da afetividade para o processamento computacional do português**. *Revista Brasileira de Linguística Aplicada*, 13(4), 2013.

[59] GIACHANOU, A.; CRESTANI, F. **Like it or not: A survey of twitter sentiment analysis methods**. *ACM Comput. Surv.*, 49(2):28:1–28:41, June 2016.

[60] GILBERT, C. H. E. **Vader: A parsimonious rule-based model for sentiment analysis of social media text**. 2014.

[61] GO, A.; BHAYANI, R.; HUANG, L. **Twitter sentiment classification using distant supervision**. *Processing*, 150, 01 2009.

[62] GONCALVES, I.; SILVA, S. **Experiments on controlling overfitting in genetic programming**. In: *Local proceedings of the 15th Portuguese Conference on Artificial Intelligence: Progress in Artificial Intelligence*, EPIA 2011, p. 152–166, Oct. 2011.

[63] GONZÁLEZ-IBÁNEZ, R.; MURESAN, S.; WACHOLDER, N. **Identifying sarcasm in twitter: a closer look**. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers-Volume 2*, p. 581–586. Association for Computational Linguistics, 2011.

[64] GOVINDARAJAN, M. **Sentiment analysis of movie reviews using hybrid method of naive bayes and genetic algorithm**. *International Journal of Advanced Computer Research*, 3(4):139, 2013.

[65] GRACZYK, M.; LASOTA, T.; TRAWIŃSKI, B.; TRAWIŃSKI, K. **Comparison of bagging, boosting and stacking ensembles applied to real estate appraisal**. In: *Asian Conference on Intelligent Information and Database Systems*, p. 340–350. Springer, 2010.

[66] GRAFF, M.; TELLEZ, E. S.; ESCALANTE, H. J.; MIRANDA-JIMÉNEZ, S. **Semantic genetic programming for sentiment analysis**. In: *NEO 2015*, p. 43–65. Springer, 2017.

[67] GRINGS, A.; OTHERS. **Regressão simbólica via programação genética: um estudo de caso com modelagem geofísica**. 2006.

[68] GUIMARAES, N.; TORGO, L.; FIGUEIRA, A. **Lexicon expansion system for domain and time oriented sentiment analysis**. In: *Proceedings of the 8th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2016)*, p. 463–471, 2016.

[69] GÜNTHER, T.; FURRER, L. **Gu-mlt-lt: Sentiment analysis of short messages using linguistic features and stochastic gradient descent**. In: *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, volume 2, p. 328–332, 2013.

[70] GÜNTHER, T.; VANCOPPENOLLE, J.; JOHANSSON, R. **Rtrgo: Enhancing the gu-mlt-lt system for sentiment analysis of short messages**. In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, p. 497–502, 2014.

[71] HADANO, M.; SHIMADA, K.; ENDO, T. **Aspect identification of sentiment sentences using a clustering algorithm**. *Procedia-Social and Behavioral Sciences*, 27:22–31, 2011.

[72] HAMDAN, H.; BÉCHET, F.; BELLOT, P. **Experiments with dbpedia, wordnet and sentiwordnet as resources for sentiment analysis in micro-blogging**. In: *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, volume 2, p. 455–459, 2013.

[73] HAMDAN, H.; BELLOT, P.; BECHET, F. **Lsislif: Crf and logistic regression for opinion target extraction and sentiment polarity analysis**. In: *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, p. 753–758, 2015.

[74] HASSAN, A.; ABBASI, A.; ZENG, D. **Twitter sentiment analysis: A bootstrap ensemble framework**. In: *Social Computing (SocialCom), 2013 International Conference on*, p. 357–364. IEEE, 2013.

[75] HASTIE, T.; TIBSHIRANI, R. **Classification by pairwise coupling**. In: *Advances in neural information processing systems*, p. 507–513, 1998.

[76] HLTCOE, J. **Semeval-2013 task 2: Sentiment analysis in twitter**. *Second Joint Conference on Lexical and Computational Semantics*, 2013.

[77] HOSMER JR, D. W.; LEMESHOW, S.; STURDIVANT, R. X. **Applied logistic regression**, volume 398. John Wiley & Sons, 2013.

[78] HSU, H.; LACHENBRUCH, P. A. **Paired t test**. *Wiley StatsRef: Statistics Reference Online*, 2014.

[79] HU, M.; LIU, B. **Mining and summarizing customer reviews**. In: *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, p. 168–177. ACM, 2004.

[80] HUSSEIN, D. M. E.-D. M. **A survey on sentiment analysis challenges**. *Journal of King Saud University-Engineering Sciences*, 2016.

[81] INHASZ, R. **Programação genética: operadores de crossover, blocos construtivos e emergência semântica**, 2010.

[82] JAGGI, M.; UZDILLI, F.; CIELIEBAK, M. **Swiss-chocolate: Sentiment detection using sparse svms and part-of-speech n-grams**. In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, p. 601–604, 2014.

[83] JAIN, V. K.; KUMAR, S. **An effective approach to track levels of influenza-a (h1n1) pandemic in india using twitter**. *Procedia Computer Science*, 70:801–807, 2015.

[84] KAJI, N.; KITSUREGAWA, M. **Building lexicon for sentiment analysis from massive collection of html documents**. In: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, p. 1075–1083, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[85] KANAYAMA, H.; NASUKAWA, T. **Fully automatic lexicon expansion for domain-oriented sentiment analysis**. In: *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, p. 355–363, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[86] KANAYAMA, H.; NASUKAWA, T. **Fully automatic lexicon expansion for domain-oriented sentiment analysis**. In: *Proceedings of the 2006 conference on empirical*

methods in natural language processing, p. 355–363. Association for Computational Linguistics, 2006.

[87] KARAMPATSIS, R.-M.; PAVLOPOULOS, J.; MALAKASIOTIS, P. **Aueb: Two stage sentiment analysis of social network messages**. In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, p. 114–118, 2014.

[88] KARANASOU, M.; AMPLA, A.; DOULKERIDIS, C.; HALKIDI, M. **Scalable and real-time sentiment analysis of twitter data**. In: *Data Mining Workshops (ICDMW), 2016 IEEE 16th International Conference on*, p. 944–951. IEEE, 2016.

[89] KESHAVARZ, H.; ABADEH, M. S. **Alga: Adaptive lexicon learning using genetic algorithm for sentiment analysis of microblogs**. *Knowledge-Based Systems*, 122:1–16, 2017.

[90] KIRITCHENKO, S.; ZHU, X.; MOHAMMAD, S. M. **Sentiment analysis of short informal texts**. *Journal of Artificial Intelligence Research*, 50:723–762, 2014.

[91] KITTLER, J.; HATEF, M.; DUIN, R. P.; MATAS, J. **On combining classifiers**. *IEEE transactions on pattern analysis and machine intelligence*, 20(3):226–239, 1998.

[92] KOULOUMPIS, E.; WILSON, T.; MOORE, J. D. **Twitter sentiment analysis: The good the bad and the omg!** 2011.

[93] KOZA, J. R. **Genetic Programming: On the Programming of Computers by Means of Natural Selection**. MIT Press, Cambridge, MA, USA, 1992.

[94] KOZA, J. R.; KEANE, M. A.; STREETER, M. J.; MYDLOWEC, W.; YU, J.; LANZA, G. **Genetic programming IV: Routine human-competitive machine intelligence**, volume 5. Springer Science & Business Media, 2006.

[95] KUNCHEVA, L. I. **Combining pattern classifiers: methods and algorithms**. John Wiley & Sons, 2004.

[96] LACERDA, A. **Uso de Programação Genética Para Propaganda Direcionada Baseada em Conteúdo**. PhD thesis, 2008.

[97] LACERDA, W. S.; DE PÁDUA BRAGA, A. **Experimento de um classificador de padrões baseado na regra naive de bayes.** *INFOCOMP*, 3(1):30–35, 2004.

[98] LANGDON, W. **Minimising testing in genetic programming**. *RN*, 11(10):1, 2011.

[99] LANGDON, W. B. **Genetic programming and data structures**. 1996.

[100] LEAL, J.; PINTO, S.; BENTO, A.; OLIVEIRA, H. G.; GOMES, P. **Cisuc-kis: tackling message polarity classification with a large and diverse set of features**. In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, p. 166–170, 2014.

[101] LEITE, V. **Uma análise da classificacao de litologias utilizando SVM, MLP e métodos Ensemble**. PhD thesis, Dissertacao de Mestrado. Departamento de Informática. Rio de Janeiro: Pontificia Universidade Católica do Rio de Janeiro, 2012.

[102] LIMA, C. A. D. M.; OTHERS. **Comitê de máquinas: uma abordagem unificada empregando máquinas de vetores-suporte**. 2004.

[103] LIN, J.; KOLCZ, A. **Large-scale machine learning at twitter**. In: *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, p. 793–804. ACM, 2012.

[104] LIU, B. **Sentiment analysis: A multifaceted problem**. *IEEE Intelligent Systems*, 25(3):76–80, 8 2010.

[105] LIU, B. **Sentiment analysis and subjectivity.** 2010.

[106] LIU, B. **Sentiment analysis and opinion mining**. *Synthesis lectures on human language technologies*, 5(1):1–167, 2012.

[107] LOPES, C. L. D. V.; SANTOS, G. H. P.; MARTINS, F. V. C. **Programação genética aplicada no processo de previsão: um estudo de caso aplicado em chamadas de uma central de teleatendimento**. *Congresso Brasileiro de Inteligência Computacional (CBIC 2017)*, 2017.

[108] LOPES, M. M. **Programação Genética para otimização de séries temporais com dados faltantes**. PhD thesis, UNIVERSIDADE FEDERAL DO RIO DE JANEIRO, 2007.

[109] LORENA, A. C.; DE CARVALHO, A. C. **Uma introdução às support vector machines**. *Revista de Informática Teórica e Aplicada*, 14(2):43–67, 2007.

[110] LORENA, A. C.; DE CARVALHO, A. C. **Estratégias para a combinaçao de classificadores binários em soluçoes multiclasses**. *Revista de Informática Teórica e Aplicada*, 15(2):65–86, 2008.

[111] LUKE, S.; BALAN, G. C.; PANAIT, L. **Population implosion in genetic programming**. In: *Genetic and Evolutionary Computation Conference*, p. 1729–1739. Springer, 2003.

[112] LUKE, S.; PANAIT, L. **A survey and comparison of tree generation algorithms**. In: *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, p. 81–88. Morgan Kaufmann Publishers Inc., 2001.

[113] LUNANDO, E.; PURWARIANTI, A. **Indonesian social media sentiment analysis with sarcasm detection**. In: *Advanced Computer Science and Information Systems (ICACSIS), 2013 International Conference on*, p. 195–198. IEEE, 2013.

[114] MALANDRAKIS, N.; FALCONE, M.; VAZ, C.; BISOGNI, J. J.; POTAMIANOS, A.; NARAYANAN, S. **Sail: Sentiment analysis using semantic similarity and contrast features**. In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, p. 512–516, 2014.

[115] MALANDRAKIS, N.; KAZEMZADEH, A.; POTAMIANOS, A.; NARAYANAN, S. **Sail: A hybrid approach to sentiment analysis.** 2013.

[116] MARQUES, L. G.; OTHERS. **Programação genética paralela com pareto: uma ferramenta para modelagem via regressão simbólica**. 2013.

[117] MARTINEZ-CÁMARA, E.; GUTIÉRREZ-VÁZQUEZ, Y.; FERNÁNDEZ, J.; MONTEJO-RÁEZ, A.; MUNOZ-GUILLENA, R. **Ensemble classifier for twitter sentiment analysis**. 2015.

[118] MATTOS, C. L. C. **Comitês de Classificadores Baseados nas Redes SOM e Fuzzy ART com Sintonia de Parâmetros e Seleção de Atributos via Metaheurísticas Evolucionárias**. PhD thesis, 2011.

[119] MAYNARD, D.; GREENWOOD, M. A. **Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis**. In: *LREC 2014 Proceedings*. ELRA, 2014.

[120] MEDHAT, W.; HASSAN, A.; KORASHY, H. **Sentiment analysis algorithms and applications: A survey**. *Ain Shams Engineering Journal*, 5(4):1093–1113, 2014.

[121] MERZ, C. J. **Using correspondence analysis to combine classifiers**. *Machine Learning*, 36(1-2):33–58, 1999.

[122] MITTAL, A.; GOEL, A. **Stock prediction using twitter sentiment analysis**. 2012.

[123] MIURA, Y.; SAKAKI, S.; HATTORI, K.; OHKUMA, T. **Teamx: A sentiment analyzer with enhanced lexicon mapping and weighting scheme for unbalanced data**. In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, p. 628–632, 2014.

[124] MOHAMMAD, S. M.; KIRITCHENKO, S.; ZHU, X. **Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets**. In: *Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013)*, Atlanta, Georgia, USA, June 2013.

[125] MOHAMMAD, S. M.; TURNEY, P. D. **Crowdsourcing a word-emotion association lexicon**. 29(3):436–465, 2013.

[126] MONTANA, D. J. **Strongly typed genetic programming**. *Evolutionary computation*, 3(2):199–230, 1995.

[127] MORAES, S. M.; SANTOS, A. L.; REDECKER, M.; MACHADO, R. M.; MENEGUZZI, F. R. **Comparing approaches to subjectivity classification: A study on portuguese tweets**. In: *International Conference on Computational Processing of the Portuguese Language*, p. 86–94. Springer, 2016.

[128] MORAGLIO, A.; KRAWIEC, K.; JOHNSON, C. G. **Geometric semantic genetic programming**. In: *International Conference on Parallel Problem Solving from Nature*, p. 21–31. Springer, 2012.

[129] MUDINAS, A.; ZHANG, D.; LEVENE, M. **Combining lexicon and learning based approaches for concept-level sentiment analysis**. In: *Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining*, p. 5. ACM, 2012.

[130] MUKHERJEE, S.; JOSHI, S. **Sentiment aggregation using conceptnet ontology**. In: *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, p. 570–578, 2013.

[131] MURPHY, K. P.; OTHERS. **Naive bayes classifiers**. *University of British Columbia*, 18, 2006.

[132] NAKOV, P.; RITTER, A.; ROSENTHAL, S.; SEBASTIANI, F.; STOYANOV, V. **Semeval-2016 task 4: Sentiment analysis in twitter.** 2016.

[133] NIELSEN, F. Å. **Afinn**, mar 2011.

[134] NIELSEN, F. Å. **A new anew: Evaluation of a word list for sentiment analysis in microblogs**. *arXiv preprint arXiv:1103.2903*, 2011.

[135] OUSSAIDENE, M.; CHOPARD, B.; PICTET, O. V.; TOMASSINI, M. **Parallel genetic programming and its application to trading model induction**. *Parallel Computing*, 23(8):1183–1198, 1997.

[136] OYEBODE, O. K.; ADEYEMO, J. A. **Genetic programming: principles, applications and opportunities for hydrological modelling**. 2014.

[137] ÖZTÜRK, N.; AYVAZ, S. **Sentiment analysis on twitter: A text mining approach to the syrian refugee crisis**. *Telematics and Informatics*, 35(1):136–147, 2018.

[138] PAGOLU, V. S.; REDDY, K. N.; PANDA, G.; MAJHI, B. **Sentiment analysis of twitter data for predicting stock market movements**. In: *Signal Processing, Communication, Power and Embedded System (SCOPES), 2016 International Conference on*, p. 1345–1350. IEEE, 2016.

[139] PAK, A.; PAROUBEK, P. **Twitter based system: Using twitter for disambiguating sentiment ambiguous adjectives**. In: *Proceedings of the 5th International Workshop on Semantic Evaluation*, p. 436–439. Association for Computational Linguistics, 2010.

[140] PANG, B.; LEE, L.; VAITHYANATHAN, S. **Thumbs up?: sentiment classification using machine learning techniques**. In: *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, p. 79–86. Association for Computational Linguistics, 2002.

[141] PANG, B.; LEE, L.; OTHERS. **Opinion mining and sentiment analysis**. *Foundations and Trends® in Information Retrieval*, 2(1–2):1–135, 2008.

[142] PATELLI, A. **Genetic programming techniques for nonlinear systems identification**. PhD thesis, Gh. Asachi Technical University of Iasi, Romania, 2011.

[143] 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*, 12:2825–2830, 2011.

[144] PINHEIRO, V. H. C. **PROGRAMAÇÃO GENÉTICA APLICADA À IDENTIFICAÇÃO DE ACIDENTES DE UMA USINA NUCLEAR PWR**. PhD thesis, Universidade Federal do Rio de Janeiro, 2018.

[145] POLI, R.; LANGDON, W.; MCPHEE, N.; KOZA, J. **A Field Guide to Genetic Programming**. 2008.

[146] POLLACK, J. B. **Parallel genetic programming and fine-grained simd architecture**. 2015.

[147] POZO, A.; ISHIDA, C.; SPINOSA, E.; RODRIGUES, E. M. **Computação evolutiva**. 2005.

[148] PRABOWO, R.; THELWALL, M. **Sentiment analysis: A combined approach**. *Journal of Informetrics*, 3(2):143–157, 2009.

[149] PRASAD, A. G.; SANJANA, S.; BHAT, S. M.; HARISH, B. **Sentiment analysis for sarcasm detection on streaming short text data**. In: *Knowledge Engineering and Applications (ICKEA), 2017 2nd International Conference on*, p. 1–5. IEEE, 2017.

[150] QIU, G.; LIU, B.; BU, J.; CHEN, C. **Opinion word expansion and target extraction through double propagation**. *Computational linguistics*, 37(1):9–27, 2011.

[151] RAMBOCAS, M.; GAMA, J. **The role of sentiment analysis**. Technical report, 2013.

[152] RAMBOCAS, M.; PACHECO, B. G. **Online sentiment analysis in marketing research: a review**. *Journal of Research in Interactive Marketing*, 2018.

[153] RIBEIRO, F. N.; ARAÚJO, M.; GONÇALVES, P.; BENEVENUTO, F.; GONÇALVES, M. A. **Sentibench-a benchmark comparison of state-of-the-practice sentiment analysis methods**. *arXiv preprint arXiv:1512.01818*, 2015.

[154] RIBEIRO, L. B. **Análise de sentimento em comentários sobre aplicativos para dispositivos móveis: estudo do impacto do pré-processamento**. 2015.

[155] RILOFF, E.; WIEBE, J. **Learning extraction patterns for subjective expressions**. In: *Proceedings of the 2003 conference on Empirical methods in natural language processing*, p. 105–112. Association for Computational Linguistics, 2003.

[156] RIZWANA, K.; KALPANA, B. **A survey on sentiment analysis and opinion mining**. 2018.

[157] RODRIGUES, E. L. M. **Evolução de funções em programação genética orientada a gramáticas**. 2010.

[158] RODRIGUES, R. G.; OTHERS. **Sentihealth-cancer: uma ferramenta de análise de sentimento para ajudar a detectar o humor de pacientes de câncer em uma rede social online**. 2016.

[159] RODRÍGUEZ-PENAGOS, C.; BATALLA, J. A.; CODINA-FILBA, J.; GARCÍA-NARBONA, D.; GRIVOLLA, J.; LAMBERT, P.; SAURÍ, R. **Fbm: Combining lexicon-based ml and heuristics for social media polarities**. In: *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, volume 2, p. 483–489, 2013.

[160] ROKACH, L. **Pattern classification using ensemble methods**, volume 75. World Scientific, 2010.

[161] ROSA, R. L. **Análise de sentimentos e afetividade de textos extraídos das redes sociais.** PhD thesis, Universidade de São Paulo, 2015.

[162] ROSENTHAL, S.; NAKOV, P.; KIRITCHENKO, S.; MOHAMMAD, S.; RITTER, A.; STOYANOV, V. **Semeval-2015 task 10: Sentiment analysis in twitter**. In: *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, p. 451–463, Denver, Colorado, June 2015. Association for Computational Linguistics.

[163] ROSENTHAL, S.; RITTER, A.; NAKOV, P.; STOYANOV, V. **Semeval-2014 task 9: Sentiment analysis in twitter**. In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, p. 73–80. Association for Computational Linguistics, 2014.

[164] SAIAS, J. **Senti. ue: Tweet overall sentiment classification approach for semeval-2014 task 9**. Association for Computational Linguistics, 2014.

[165] SAIF, H.; FERNANDEZ, M.; HE, Y.; ALANI, H. **Evaluation datasets for twitter sentiment analysis: a survey and a new dataset, the sts-gold**. 2013.

[166] SALMI, M. H. S. **Investigação de algoritmos de análise de sentimento para a língua portuguesa**, 2015.

[167] SALUNKHE, P.; SURNAR, A.; SONAWANE, S. **A review: Prediction of election using twitter sentiment analysis**. *International Journal of Advanced Research in Computer Engineering Technology*, 2017.

[168] SANTORINI, B. **Part-of-speech tagging guidelines for the penn treebank project (3rd revision)**. *Technical Reports (CIS)*, p. 570, 1990.

[169] SARKAR, D.; BALI, R.; SHARMA, T. **Practical machine learning with python: A problem-solver's guide to building real-world intelligent systems**. 2017.

[170] SCHAPIRE, R. E. **A brief introduction to boosting**. 1999.

[171] SCHUMAKER, R. P.; CHEN, H. **Textual analysis of stock market prediction using breaking financial news: The azfin text system**. *ACM Transactions on Information Systems (TOIS)*, 27(2):12, 2009.

[172] SHARIF, W.; SAMSUDIN, N. A.; DERIS, M. M.; NASEEM, R. **Effect of negation in sentiment analysis**. In: *Innovative Computing Technology (INTECH), 2016 Sixth International Conference on*, p. 718–723. IEEE, 2016.

[173] SILVA, J. R. D.; OTHERS. **Detecção de opiniões e análise de polaridade em documentos financeiros com múltiplas entidades.** 2015.

[174] SILVA, L. L. A. A. **Análise de sentimentos em contexto: estudo de caso em blog de empreendedorismo**. 2015.

[175] SILVA, N. F. F. D. **Análise de sentimentos em textos curtos provenientes de redes sociais**. PhD thesis, Universidade de São Paulo, 2016.

[176] SOUSA, R. T.; OTHERS. **Avaliação de classificadores na classificação de radiografias de tórax para o diagnóstico de pneumonia infantil**. 2013.

[177] SPERIOSU, M.; SUDAN, N.; UPADHYAY, S.; BALDRIDGE, J. **Twitter polarity classification with label propagation over lexical links and the follower graph**. In: *Proceedings of the First workshop on Unsupervised Learning in NLP*, p. 53–63. Association for Computational Linguistics, 2011.

[178] STEINWART, I.; CHRISTMANN, A. **Support vector machines**. Springer Science & Business Media, 2008.

[179] STONE, P. J.; DUNPHY, D. C.; SMITH, M. S. **The general inquirer: A computer approach to content analysis.** 1966.

[180] TABOADA, M.; BROOKE, J.; TOFILOSKI, M.; VOLL, K.; STEDE, M. **Lexicon-based methods for sentiment analysis**. *Computational linguistics*, 37(2):267–307, 2011.

[181] TANG, D.; WEI, F.; QIN, B.; LIU, T.; ZHOU, M. **Coooolll: A deep learning system for twitter sentiment classification**. In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, p. 208–212, 2014.

[182] TURNEY, P. D. **Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews**. In: *Proceedings of the 40th annual meeting on association for computational linguistics*, p. 417–424. Association for Computational Linguistics, 2002.

[183] TURNEY, P. D.; LITTMAN, M. L. **Unsupervised learning of semantic orientation from a hundred-billion-word corpus**. *arXiv preprint cs/0212012*, 2002.

[184] VELICHKOV, B.; KAPUKARANOV, B.; GROZEV, I.; KARANESHEVA, J.; MIHAYLOV, T.; KIPROV, Y.; NAKOV, P.; KOYCHEV, I.; GEORGIEV, G. **Su-fmi: System description for semeval-2014 task 9 on sentiment analysis in twitter**. In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, p. 590–595, 2014.

[185] VOHRA, S.; TERAIYA, J. **A comparative study of sentiment analysis techniques**. *Journal JIKRCE*, 2(2):313–317, 2013.

[186] VRAJITORU, D. **Large population or many generations for genetic algorithms? implications in information retrieval**. In: *Soft Computing in Information Retrieval*, p. 199–222. Springer, 2000.

[187] WANI, G. P.; ALONE, N. V. **Analysis of indian election using twitter**. *International Journal of Computer Applications*, 121(22), 2015.

[188] WIJKSGATAN, O.; FURRER, L. **Gu-mlt-lt: Sentiment analysis of short messages using linguistic features and stochastic gradient descent**. *Atlanta, Georgia, USA*, 328, 2013.

[189] WILSON, T.; WIEBE, J.; HOFFMANN, P. **Recognizing contextual polarity in phrase-level sentiment analysis**. In: *Proceedings of the conference on human language technology and empirical methods in natural language processing*, p. 347–354. Association for Computational Linguistics, 2005.

[190] WOLPERT, D. H. **Stacked generalization**. *Neural networks*, 5(2):241–259, 1992.

[191] WU, L.; MORSTATTER, F.; LIU, H. **Slangsd: Building and using a sentiment dictionary of slang words for short-text sentiment classification**. *CoRR*, abs/1608.05129, 2016.

[192] YANG, S. Y.; MO, S. Y. K.; LIU, A.; KIRILENKO, A. A. **Genetic programming optimization for a sentiment feedback strength based trading strategy**. *Neurocomputing*, 264:29–41, 2017.

[193] ZHANG, L.; XU, W.; LI, S. **Aspect identification and sentiment analysis based on nlp**. In: *Network Infrastructure and Digital Content (IC-NIDC), 2012 3rd IEEE International Conference on*, p. 660–664. IEEE, 2012.

[194] ZHOU, Z.-H. **Ensemble methods: foundations and algorithms**. Chapman and Hall/CRC, 2012.

[195] ZHOU, Z.; ZHANG, X.; SANDERSON, M. **Sentiment analysis on twitter through topic-based lexicon expansion**. In: *Australasian Database Conference*, p. 98–109. Springer, 2014.

[196] ZHU, X.; KIRITCHENKO, S.; MOHAMMAD, S. **Nrc-canada-2014: Recent improvements in the sentiment analysis of tweets**. In: *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, p. 443–447, 2014.

[197] ZIMBRA, D.; ABBASI, A.; ZENG, D.; CHEN, H. **The state-of-the-art in twitter sentiment analysis: A review and benchmark evaluation**. 2010.