Aplicando Programação Genética na Mineração de Opiniões

Airton Bordin Junior
Instituto de Informática – Universidade Federal de Goiás (UFG)
Caixa Postal 131 – 74690-900 – Goiânia – GO – Brazil
Email: airtonbjunior@gmail.com

***Abstract*—Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.**

***Keywords*—Análise de Sentimentos, Mineração de Opiniões, Programação Genética, Classificadores.**

I. Introdução

A Análise de Sentimentos (AS) é uma linha de pesquisa que tem por objetivo a classificação das emoções de um determinado texto, geralmente como positivo, negativo ou neutro [1]. 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] e Facebook[2], e a forma como os usuários compartilham suas opiniões e sentimentos sobre os mais diversos assuntos, tem motivado a pesquisa de classificadores para esses conteúdos.

As abordagens de classificação de sentimentos são comumente divididas em três classes principais [2], [3], [4]: técnicas utilizando Aprendizado de Máquina, baseadas em Léxico e Híbridas. A primeira delas utiliza abordagens de Aprendizado de Máquina supervisionado para a classificação das opiniões, realizando o treinamento com mensagens previamente classificadas. As abordagens Léxicas são heurísticas criadas manualmente, baseadas em aspectos estruturais do texto e fazem uso de Dicionários Léxicos - conjunto de palavras e sua polaridade. As técnicas híbridas fazem uso das duas abordagens anteriores de forma conjunta.

O presente trabalho é uma evolução da pesquisa publicada em [5] [...]

[Continuar introdução]

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

II. Conceitos

A. Análise de Sentimentos

A Análise de Sentimentos - também chamada de Análise de Opiniões ou Mineração de Opiniões - é uma linha de pesquisa abrangente e que vem sendo tema de diversos trabalhos nos últimos anos. Como observado em [1], 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.

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, negativa ou neutra.

A AS pode ser classificada em 4 categorias específicas [2]:

- AS em nível de Documento: assume-se que um documento contém uma opinião principal sobre um determinado assunto expressa pelo autor;
- AS em nível de Sentença: considera que há uma única opinião sobre determinado assunto na sentença;
- AS em nível de Aspecto: leva em consideração a opinião sobre diversos aspectos de uma mesma entidade;
- AS Comparativa: a opinião não é expressa de forma direta e sim de forma comparativa a outra entidade.

Além disso, as abordagens de AS podem ser divididas em soluções utilizando Aprendizado de Máquina Supervisionado, Baseadas em Léxico e abordagem híbrida [2], [6].

Em soluções supervisionadas, técnicas de Aprendizado de Máquina (AM) são aplicados à mensagens previamente rotuladas de forma a identificar características que auxiliem na distinção e detecção de sentimentos nas sentenças desconhecidas. Dentre as principais técnicas de AM para a classificação de sentimentos, podemos citar o *Support Vector Machines* (SVM), *Naïve Bayes*, *Adaboost*, Redes Neurais Artificiais, entre outros.

Técnicas baseadas em Léxicos atuam principalmente em características sintáticas e semânticas do texto e fazem uso de Dicionários Léxicos - conjunto de palavras e suas polaridades (grau de positividade e negatividade de uma mensagem). À partir desse Dicionário, é feito o processamento das mensagens pelo classificador e retornada a polaridade das mesmas [7].

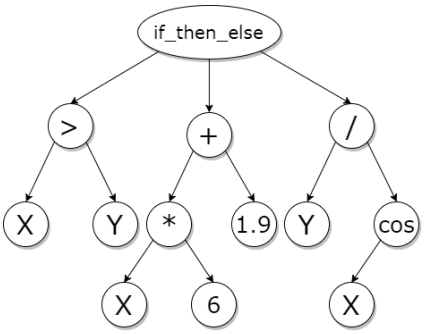

Fig. 1: Exemplo de um programa em Programação Genética

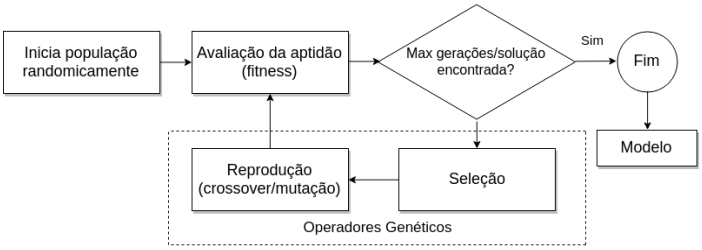

Fig. 2: Fluxo geral da PG

Abordagens híbridas fazem uso tanto de técnicas de AM quanto baseadas em Léxico para a classificação das mensagens e geralmente os Dicionários Léxicos tem papel central nesse processo [6].

O presente trabalho faz uma análise de sentimentos em nível de sentença, ou seja, considera que cada mensagem exprime uma opinião sobre determinado assunto e não considera diferentes aspectos do mesmo. Em trabalhos de AS analisando *tweets* essa é a abordagem mais comum, principalmente pela limitação na quantidade de caracteres das mensagens. Além disso, usa uma abordagem híbrida para o processo de classificação das mensagens, fazendo uso de técnicas utilizando Dicionários Léxicos e Aprendizado de Máquina.

B. Programação Genética

Programação Genética (PG) é um campo da computação evolucionária que busca resolver problemas, de forma automatizada, sem demandar conhecimentos detalhados sobre a solução [8]. De forma geral, podemos 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, à partir de uma ou mais entradas, produzir uma saída para as mesmas. Embora possam ser representadas por diversos tipos de estruturas, a forma mais comum é a representação por meio de árvores, onde os nós internos representam funções e os nós folha representam terminais do problema. Um exemplo de programa pode ser visto na Figura 1, que representa o código *if (X >Y) then { X * 6 + 1.9 } else { X / cos(X) }*.

Na PG, assim como em outros algoritmos baseados na evolução humana, são criadas populações onde cada indivíduo representa uma possível solução para o problema. A inicialização aleatória é a forma mais comum de criação da população, evoluindo as mesmas no decorrer dos ciclos, chamados de gerações. A cada geração, indivíduos possivelmente melhores são criados, evoluindo os programas (modelos) gerados. 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 [9]. Um fluxo geral do funcionamento de um algoritmo de PG pode ser visto na fig 2.

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 escolha de uma boa função *fitness* está intimamente ligada ao tipo do problema.

Os responsáveis pela evolução da população de indivíduos são os operadores genéticos. Para a Programação Genética, os principais operadores são a seleção, mutação e *crossover*. Na seleção, um indivíduo é escolhido para fazer parte da próxima geração. A mutação modifica um nó da árvore, de forma a criar um indivíduo modificado em uma das partes escolhida aleatoriamente. O *crossover* realiza o cruzamento entre dois indivíduos (pais), gerando duas novas possíveis soluções para o problema (filhos). Além desses principais operadores, existem outros como a edição, encapsulamento e a destruição [10].

Detalhes dos parâmetros gerais da PG utilizados neste trabalho podem ser vistos na tabela V.

III. Trabalhos relacionados

O número de pesquisas na área de Análise de Sentimentos vem crescendo a cada ano, motivado, principalmente, pela importância da área no contexto atual de análise de grande quantidade de dados e informações.

A formalização matemática de opinião como uma quíntupla (entidade, aspecto da entidade, sentimento, autor e tempo), definida por [1], é utilizada em grande parte dos trabalhos na área, caracterizando-se, portanto, como elemento fundamental nas pesquisas sobre o assunto.

Em [11] os autores apresentam uma abordagem de um classificador linear treinado utilizando *Stochastic Gradient Descent* (SGD). Além do pré-processamento das entradas, faz uso de características das mensagens, como a soma acumulada de polaridades positivas e negativas, usando o dicionário SentiWordNet[3] e o *stem*[4] da frase. Além disso, o classificador trabalha com 3 variantes de cada palavra: 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 um F1-score de 65.54 no *benchmark* SemEval 2013.

[3] http://sentiwordnet.isti.cnr.it/

[4] Processo de redução de uma flexionada à sua raíz.

A maior parte das pesquisas usa abordagens baseadas em Aprendizado de Máquina Supervisionado ou baseadas em Léxicos. Entretanto, alguns trabalhos vem apresentando resultados promissores com a utilização de abordagens híbridas, com os Dicionários Léxicos possuindo papel central no processo de AS [6].

Em [12] é utilizada uma abordagem híbrida para a AS de *tweets*. Para a Análise Léxica, são utilizados os dicionários SentiStrength[5] e LIU[6], além de um dicionário de palavras de negação construído manualmente. A definição da polaridade das mensagens nessa abordagem é feita pela soma simples de polaridade das palavras, atribuindo o valor +1 para palavras positivas e -1 para negativas. Caso o resultado do processo de Análise Léxica não atinja um *threshold* de valroes definido, o processo de AS é feito por um módulo de AM utilizando SVM. O trabalho obteve um F1-score de 63.94 para a base de teste de *Tweets2014* do *benchmark* SemEval 2014.

Combinação de abordagem Léxica e Aprendizado de Máquina para fazer a AS em *reviews* de *softwares*[7] e filmes[8] pode ser vista em [13]. Nesse trabalho, é utilizado um Dicionário Léxico próprio manualmente anotado com 7048 palavras com polaridades variando entre -3 e 3. Além disso, considera para a análise sintática palavras de negação e verbos e adjetivos de comparação. Para a análise via AM faz uso de SVM utilizando como *features* a polaridade das palavras, palavras de negação e o *score* retornado pelo módulo de Análise Léxica. A pesquisa obteve uma acurácia de 89.64% em *reviews* de *softwares* e 82.30% de vídeos utilizando a abordagem híbrida.

Uma análise híbrida iniciando com a avaliação prévia de *tweets* utilizando *emoticons* pode ser visto em [14]. Em um primeiro momento, é feito um filtro das mensagens de acordo com um dicionário de *emoticons* positivos ou negativos em suas respectivas classes. Após a classificação inicial, o trabalho utiliza uma abordagem de Aprendizado de Máquina usando Naïve Bayes. Como *features* usa um conjunto de *n-grams* pré-processadas, com a remoção de *stopwords*[9], URL e menções a outros usuários do *Twitter*. O trabalho atingiu uma acurácia de 64% na avaliação das mensagens.

Uma proposta de combinação de análise baseada em regras léxicas e Aprendizado de Máquina é apresentada em [15]. O trabalho avalia *reviews* de produtos, filmes e mensagens do MySpace[10]. Faz uso do Dicionário Léxico *General Inquirer*[11], contendo 3672 palavras classificadas como positivas ou negativas e regras gramaticais para a avaliação das mensagens. Para a análise utilizando AM faz uso de SVM usando como método de seleção de *features* o *document frequency* [16]. O processo híbrido é feito criando uma aplicação dos classificadores em sequência.

Em [17], o autor descreve sua solução híbrida para AS. Um modelo de classificação baseada em Léxico combinado com uma abordagem fazendo uso de *maximum entropy* com bigramas. O Dicionário Léxico utilizado foi obtido de forma automatizada à partir de um conjunto de documentos coletados da web e utilizando o algoritmo de [18] para a definição das polaridades das palavras. A fusão das duas abordagens é feita utilizando SVM e usando como *features* o F1-score de cada uma das classes (positivo, negativo e neutro) de cada um dos classificadores. O projeto obteve a 9ª colocação entre 35 participantes do SemEval 2013.

O principal diferencial do presente trabalho em relação aos artigos supracitados, principalmente os de abordagem híbrida, é a capacidade de gerar automaticamente um modelo léxico de AS baseado na Programação Genética. Vale destacar que essa abordagem permite uma customização ou generalização - dependendo da base de treinamento e as funções utilizadas - e também um entendimento de como o modelo atribui classe para as mensagens.

IV. ABORDAGEM

O desafio de gerar o classificador de sentimentos pode ser descrito como um problema de otimização, com o objetivo de encontrar um modelo que represente a solução desejada.

Considere M o conjunto de modelos possíveis, sendo que cada modelo m é composto de elementos $e \in E$ que analisam aspectos da mensagem sob classificação. Sendo $f_m(x)$ a função de avaliação de cada modelo m, o objetivo da otimização é encontrar um modelo m' tal que sua função $f_{m'}(x) > f_m(x) \ \forall \ f_m(x)$, maximizando o resultado dos modelos.

A métrica principal utilizada para a avaliação dos modelos e para o *fitness* da PG foi o F1-score médio das classes positivas e negativas [19]:

$$F_1^{PN} = \frac{F_1^P + F_1^N}{2} \tag{1}$$

As classes disponíveis são:

$$X = \{Positivo, Negativo, Neutro\} \tag{2}$$

O F1 de uma classe $x \in X$ é calculado como a média harmônica entre Precisão (π) e *Recall* (ρ) dessa classe:

$$F_1^x = \frac{2\pi^x \rho^x}{\pi^x + \rho^x} \tag{3}$$

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

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

$$\rho^x = \frac{PP}{PP + UP + NP} \tag{5}$$

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

[5]http://sentistrength.wlv.ac.uk/
[6]https://www.cs.uic.edu/ liub/FBS/sentiment-analysis.html#lexicon
[7]https://www.cnet.com/topics/software/products/
[8]http://www.imdb.com/
[9]Palavras que podem ser consideradas irrelevantes para a AS.
[10]https://myspace.com/
[11]http://www.wjh.harvard.edu/ inquirer/homecat.htm

TABLE I: Matriz de confusão

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

V. Experimentos

Para validar a proposta, os experimentos seguem o fluxo mostrado na Figura 3.

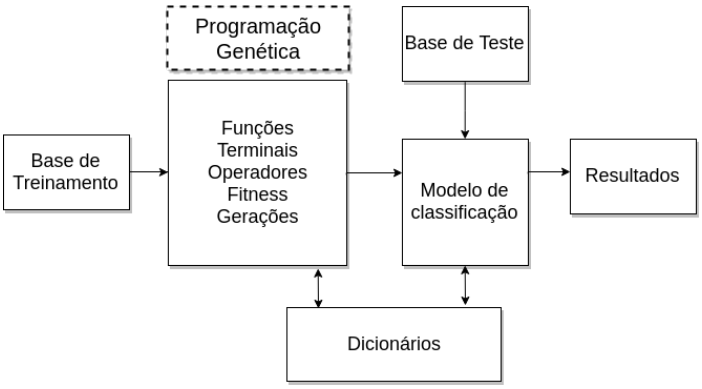

Fig. 3: Fluxo geral da Solução

Para apoiar o desenvolvimento de todos os operadores e recursos da PG foi utilizada a biblioteca DEAP[12] (*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 [20].

A. Benchmark

Como *benchmark* foi utilizada a base fornecida pelo evento SemEval 2014[13] (*International Workshop on Semantic Evaluation*), uma das principais competições na área de Processamento de Linguagem Natual (PNL).

O evento é dividido por tarefas (*Tasks*), que possuem objetivos distintos dentro da área de PNL. Para este trabalho, utilizou-se a base de dados da *Task 9 - Sentiment Analysis in Twitter*. São disponibilizadas bases de treinamento e de testes para *download*[14] no site do evento.

A base de treinamento aplicada no trabalho possui 9684 mensagens, conforme apresentado na tabela II.

O evento também disponibiliza uma base de teste com 8987 mensagens, que serve como critério de avaliação e comparação dos trabalhos submetidos para cada *Task*. A base fornecida é dividida em 5 sub-bases, como apresentado na tabela III.

[12]https://github.com/DEAP/deap

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

[14]http://alt.qcri.org/semeval2014/task9/index.php?id=data-and-tools

TABLE II: Mensagens de treinamento

Polaridade	Mensagens
Positiva	3640
Negativa	1458
Neutra	4586
TOTAL	9684

TABLE III: Mensagens de teste

Bases	Mensagens
Tweets2013	3813
Tweets2014	1853
Sarcasm	86
SMS2013	2093
LiveJournal	1142
Total	8987

As mensagens são classificadas em positivas, negativas ou neutras. Para a criação do *ranking* dos trabalhos submetidos, leva-se em consideração a média aritmética do F1 das mensagens positivas e negativas.

Para este trabalho, considerando um conjunto M de modelos m, cada *tweet* $t \in T$ será classificado em uma das 3 classes, conforme a regra apresentada:

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

B. Dicionários

Em [5], trabalho utilizado como *baseline*, utilizou-se somente o dicionário de palavras positivas e negativas de LIU, além de um dicionário de *emoticons* e *hashtags*. De forma a buscar aumentar a eficiência do classificador, novos dicionários foram incluídos no sistema.

A inclusão de novos dicionários é muito importante para a evolução do projeto, uma vez que os modelos dependem desses recursos para buscar as polaridades das palavras para avaliação. A escolha dos dicionários foi feita com base em pesquisas na literatura e todos os dicionários são disponibilizados gratuitamente para *download*.

Os dicionários utilizados e a quantidade de palavras podem ser vistos na tabela IV.

TABLE IV: Dicionários

		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	6063	5035	11098
SemEval2015	600	330	930
TODOS	43583	81643	125226

C. Parâmetros da PG

Os parâmetros gerais da PG utilizadas no trabalho são apresentados na tabela V.

TABLE V: Parâmetros gerais da PG

Parâmetro	Valor
Cruzamento	
Tipo	*One-point*
Probabilidade	90%
Mutação	
Tipo	Uniforme
Probabilidade	10%
Seleção	Torneio
Criação	*Half-and-half*
Fitness	F1-score
Elitismo	Sim

Além dos parâmetros apresentados, a definição do tamanho da população e quantidade de gerações são muito importantes para a PG. Por sua importância para os testes deste trabalho, esses parâmetros são discutidos separadamente na seção V-G.

D. Inclusão de novos dicionários

A primeira modificação em relação ao *baseline* [5] foi a inclusão de novos dicionários, listados na tabela IV.

Nessa versão inicial, todos os dicionários possuiam a mesma importância, ou seja, o mesmo peso no sistema de classificação. No caso de uma palavra ser encontrada em mais de um dicionário, uma média aritmética simples era realizada, e o resultado retornado pelo modelo.

E. Ponderação dos dicionários

De forma a buscar melhorar os resultados obtidos com a inclusão dos dicionários, pesos foram atribuídos aos mesmos.

Para cada um dos dicionários foi criado um atributo para representar seu peso na avaliação das mensagens, como pode ser visto na tabela VI.

Inicialmente, foram definidos alguns valores reais fixos para a utilização pela PG na definição dos pesos dos dicionários, conforme apresentado em (6).

$$w_i \in W \mid W = \{0.0,\ 0.5,\ 1.0,\ 1.5,\ 2.0\} \tag{6}$$

De certa forma, ao atribuírmos esses valores, perdemos a oportunidade de deixar que a própria PG identifique pesos virtualmente melhores em uma faixa real de valores. Pensando nisso, no marco posterior à definição dos pesos mostrados em (6), esses valores foram modificados para aceitar uma faixa real de pesos, conforme apresentado em (7).

$$w_i \in W \mid 0 \leq W \leq 2 \tag{7}$$

TABLE VI: Dicionários e seus pesos

Dicionário	Peso
LIU	w_1
Sentiwordnet	w_2
AFINN	w_3
Vader	w_4
Slang	w_5
Effect	w_6
SemEval2015	w_7

F. Mutação exclusiva para os valores dos pesos

Os pesos atribuídos aos dicionários nas funções da PG só poderiam ser modificados por meio do operador de mutação. Para a criação desses testes, a probabilidade de qualquer parte da árvore passar por mutação era de 10%, ou seja, para que um dos nós que representa os pesos fosse modificada, a chance seria ainda menor, uma vez que ela compartilharia dessa probabilidade com todos os outros nós da árvore (terminais e não terminais). Com isso, os valores dos pesos estavam intimamente ligados aos atribuídos durante sua criação - na primeira geração.

Como forma de facilitar a modificação desses pesos para a criação de possíveis indivíduos melhores, foi estabelecida uma mutação especial para os valores reais presentes na árvore. Essa mutação acontece de forma independente da mutação original da PG e, inclusive, podem ocorrer simultaneamente.

G. população e gerações

Dentre os principais parâmetros da PG podemos destacar o tamanho da população e a quantidade de gerações de indivíduos. [21] afirma que a configuração mais comum é a utilização de uma quantidade pequena de gerações, usualmente 51, e população variando entre 500 e 2000. A razão para esses valores é, principalmente, cultural, uma vez que foram as configurações definidas por Koza em [8].

Em nossos testes, notou-se que os melhores indivíduos estavam sendo obtidos em gerações muito próximas do limite de 51 gerações, o que reforçou a hipótese de que um aumento na quantidade de gerações poderia acarretar em modelos melhores. Um exemplo de evolução de 3 modelos utilizando a configuração supracitada (51 gerações) pode ser visto na fig 4.

Como forma de padronizar os treinamentos dos modelos, foi mantida a quantidade de ciclos de processamento em aproximadamente 25000 (50 gerações x 500 população). Os ciclos utilizados podem ser visualizados na tabela VII.

TABLE VII: Parâmetros de População e Gerações da PG

População	Gerações
500	51
250	100
170	150

H. Faixa de valores da classe neutra

Como demonstrado na seção V-A, as mensagens são classificadas como neutras somente quando o modelo retorna o

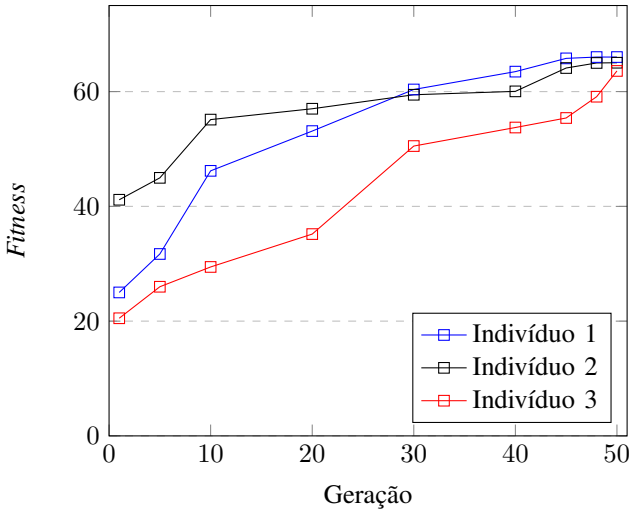

Fig. 4: Exemplo de evolução do *fitness* de 3 modelos

valor zero. Com o aumento da quantidade de dicionários e o consequente incremento no número de palavras disponíveis, houve um aumento significativo na quantidade de mensagens neutras sendo classificadas como positivas ou negativas de forma incorreta.

Uma hipótese seria permitir que a faixa de valores que caracteriza as mensagens neutras fossem definidas pela própria PG, durante a evolução dos modelos. Dessa forma, foi incluída no conjunto de funções da PG uma função para determinação do *range* dos valores da classe neutra. A fórmula de classificação das mensagens apresentada na seção V-A, portanto, foi modificada.

Sendo o Limite Superior do valor neutro identificado por SR e o Limite Inferior por IR, $IR, SR \in \mathbb{R}$, o processo de classificação dos *tweets* t pelos modelos m se dá por:

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

I. Funções e terminais da PG

A representação dos indivíduos da PG serão constituídos pela combinação de funções e terminais adequados ao domínio do problema. Como o presente trabalho possui funções com tipos de entrada e retorno diferentes, é utilizado um tipo especial de PG chamada de Programação Genética Fortemente Tipada (PGFT). Na PGFT há uma camada adicional de verificação em cada geração de indivíduos para a criação de modelos válidos.

Temos, portanto, dois conjuntos principais: o conjunto F, contendo as funções da PG e o conjunto T com os terminais. As principais funções são apresentadas na tabela VIII e os terminais em 8.

$$T = \{tweet, \delta\}, -2 \leq \delta \leq 2 \quad (8)$$

TABLE VIII: Principais funções da PG

ID	Função	Descrição
1	polSum(msg)	Soma das polaridades das palavras
2	polSumAVG(msg)	Média aritmética das polaridades
3	polSumAVGW(msg, $[w_1 ... w_7]$)	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	if_then_else(bool, c1, c2)	Se bool é true então c1 senão c2
9	removeStopwords(msg)	Remove stopwords da mensagem
10	neutralRange(il, sl)	Limite inferior/superior para classe neutra
11	add, sub, mul, div, sen, cos, exp	Funções matemáticas

J. Restrições

Métodos de penalização são as formas mais comuns de restrições em Algoritmos Evolucionários [22].

No presente trabalho, aplicamos algumas penalizações quanto à repetição de funções específicas na árvore, bem como a valores de parâmetros das funções.

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

A repetição de algumas funções na árvore que representa cada indivíduo não traz resultados melhores e, algumas vezes, causa inconsistências na definição de alguns atributos. Além disso, funções como a 1, 2 e 3 da tabela VIII são massivas e consomem muito processamento, pois iteram sobre todas as mensagens. Por esse motivo, indivíduos que possuem essas funções massivas repetidas tem seu *fitness* penalizado

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

K. Versões

De forma a avaliar cada uma das modificações no processo e facilitar a comparação dos resultados, foram definidas versões para cada um dos marcos principais de funcionalidades apresentadas. As versões e suas identificações podem ser vistas na tabela IX. Na seção de resultados, cada uma das etapas será referenciada pelo seu número de versão.

TABLE IX: Versões do sistema e suas descrições

Versão	Descrição
1.0	Versão *baseline*, publicada em [5]
2.0	Inclusão de dicionários (tabela IV)
2.1	Ponderação de dicionários com valores discretos
2.2	Ponderação de dicionários com valores reais
2.3	Mutação especial para os pesos dos dicionários
2.4	Modificação dos parâmetros de população e gerações (250p 101g)
2.5	Modificação dos parâmetros de população e gerações (170p 150g)
2.6	Faixa de valores da classe Neutra variável

VI. Resultados

Como discutido nas seções anteriores, iremos considerar como *baseline* os resultados obtidos na versão 1 da solução (conforme apresentado na tabela IX).

Para cada uma das configurações, foram gerados 30* modelos e calculados os valores médios e os melhores valores. Ainda, de forma a identificar a diferença entre os modelos gerados, o desvio padrão foi calculado para cada uma das versões, como pode ser visto na tabela X. Os valores mostram que os modelos gerados não possuem diferenças significativas, permitindo que sejam usados os melhores valores para a classificação das mensagens.

TABLE X: Desvio padrão dos modelos

Versão	Modelos	Desvio Padrão
2.0		0.12
2.1		0.64
2.2		0.26
2.3	30*	0.04
2.4		0.05
2.5		0.01
2.6		0.29

Em comparação com os resultados obtidos no *baseline*, houve uma melhora significativa nos resultados, como pode ser visto na tabela XI. Destaque para um ganho expressivo na base de Sarcasm e para as bases de Twiiter2014 e SMS. A base de LiveJournal, que havia obtido o melhor resultado no *baseline* teve um crescimento de 13%. Ainda, destaca-se que houve melhorias em todas as bases, sendo de 21% para todos as mensagens.

TABLE XI: Ganhos em relação ao *baseline*

Base	Ganhos
Twitter2013	18%
Twitter2014	32%
Sarcasm	120%
SMS	32%
LiveJournal	13%
TODOS	21%

Em comparação com os dos trabalhos submetidos para SemEval 2014, podemos identificar avanços em relação à versão anterior. Para a base LiveJournal, por exemplo, o melhor resultado obtido é somente 6.41 pontos inferior ao primeiro colocado, o que mostra que o modelo é competitivo. A maior diferença encontrada foi para a base de Sarcasm, ficando 15.05 pontos abaixo do primeiro colocado, mesmo com os ganhos obtidos nessa versão.

A tabela XIII apresenta os resultados detalhados em cada uma das versões para cada métrica utilizada. Percebe-se que 4 de 5 melhores resultados foram obtidos nas duas últimas versões, o que mostra que as hipóteses levantadas para as modificações mostraram-se válidas. Destaque para a última versão, que incluiu o valor variável para a classe neutra, com 3 melhores resultados de 5 no total.

O resultado geral resumido é apresentado na tabela XIII. Como citado anteriormente, podemos notar que os piores resultados para todas as bases são obtidos na versão de *baseline*, o que mostra a eficiéncia das modificações.

TABLE XII: Comparação de resultados com o SemEval 2014

Base	Melhor F1	Top 3 SemEval
Tweets2013	61.43	1º 72.12
		2º 70.75
		3º 70.40
Tweets2014	59.72	1º 70.96
		2º 70.14
		3º 69.95
Sarcasm	43.11	1º 58.16
		2º 57.26
		3º 56.50
SMS2013	60.06	1º 70.28
		2º 67.68
		3º 67.51
LiveJournal	68.43	1º 74.84
		2º 74.46
		3º 73.99

Um resultado relevante e que merece discussão é a atribuição dos pesos aos dicionários pelos modelos. Em todos os modelos gerados para todas as versões, os dois últimos dicionários (w_6 e w_7, SemEval2015 e Effect, respectivamente) receberam o valor 0. Isso significa que, para a configuração deste trabalho e para o *benchmark* utilizado, a PG decidiu não utilizar os mesmos.

TABLE XIV: Principais resultados

		Melhores valores						
	V1[5]	**V2.0**	**V2.1**	**V2.2**	**V2.3**	**V2.4**	**V2.5**	**V2.6**
Twitter2013	51.93	57.12	60.87	**61.43**	60.81	60.73	60.73	60.78
Twitter2014	45.07	55.33	59.56	59.47	59.67	59.35	**59.72**	59.35
Sarcasm	24.12	**53.07**	41.61	39.74	39.74	40	39.74	43.11
SMS	45.49	45.83	56.97	57.12	56.92	57.14	56.92	**60.06**
LiveJournal	60.52	62.75	67.86	68.26	**68.43**	68.41	**68.43**	**68.43**
TODAS	50.54	55.55	61.01	61.05	61.02	60.97	61.01	**61.24**

[Ver se falta discutir mais algum resultado aqui]

VII. Conclusão

Considerando a importância dos modelos de classificação de sentimento e o custo para gerá-los manualmente, o presente trabalho propõe o uso de PG para automatizar a geração desses modelos.

Para validação da abordagem, foi dado continuidade ao trabalho publicado em [5] que serviu como *baseline* para esta pesquisa.

Foram desenvolvidas melhorias progressivas, de forma a testar sua eficácia em relação aos resultados. Os treinos e testes utilizaram o mesmo *benchmark* do *baseline*, conforme apresentado nas tabelas II e III.

Percebe-se que em todas as modificações do trabalho houve melhorias em relação ao *basiline*, com destaque para última versão, com 3 melhores valores de 5 no total, o que mostra

TABLE XIII: Resultados detalhados por versão

Versão	Base	Acurácia	Média Precisão	Média Recall	F1 P/U/N	F1 P/N
2.0	Twitter2013	54.34	57.12	56.75	51.03	57.12
	Twitter2014	56.88	53.87	55.35	49.49	55.33
	Sarcasm	**52.33**	**69.47**	46.09	**44.27**	**53.07**
	SMS	32.3	49.71	49.47	30.99	45.83
	LiveJournal	52.71	61.41	53.88	45.17	62.75
	Todas	49.5	54.74	53.51	45.85	55.55
2.1	Twitter2013	61.5	60.05	62.11	59.41	60.87
	Twitter2014	60.98	55.62	60.39	56.54	59.56
	Sarcasm	45.35	52.08	53.87	42.56	41.61
	SMS	62.26	60.9	64.03	60.38	56.97
	LiveJournal	66.81	67.1	66.31	66.6	67.86
	Todas	62.07	60.68	62.1	60.52	61.01
2.2	Twitter2013	**62.63**	**61.05**	**63.44**	**60.9**	**61.43**
	Twitter2014	60.93	55.56	60.34	56.45	59.47
	Sarcasm	43.02	47.83	50.3	40.45	39.74
	SMS	62.3	60.91	64.14	60.48	57.12
	LiveJournal	67.08	67.3	66.64	66.89	68.26
	Todas	62.07	60.67	62.17	60.54	61.05
2.3	Twitter2013	61.45	60.01	62.17	59.39	60.81
	Twitter2014	61.04	**55.86**	**60.54**	56.59	59.67
	Sarcasm	43.02	47.83	50.3	40.45	39.74
	SMS	62.26	60.82	64.02	60.37	56.92
	LiveJournal	**67.16**	**67.42**	**66.74**	**67.01**	**68.43**
	Todas	62.07	60.66	62.16	60.54	61.02
2.4	Twitter2013	61.45	59.99	62.03	59.34	60.73
	Twitter2014	60.87	55.45	60.18	56.34	59.35
	Sarcasm	44.19	51.36	52.86	41.82	40
	SMS	62.35	60.91	64.19	60.52	57.14
	LiveJournal	**67.16**	67.39	**66.74**	67	68.41
	Todas	62.05	60.63	62.12	60.5	60.97
2.5	Twitter2013	61.45	59.99	62.03	59.34	60.73
	Twitter2014	**61.14**	55.72	60.48	**56.62**	**59.72**
	Sarcasm	43.02	47.83	50.3	40.45	39.74
	SMS	62.26	60.82	64.02	60.37	56.92
	LiveJournal	**67.16**	**67.42**	**66.74**	**67.01**	**68.43**
	Todas	62.07	60.66	62.17	60.53	60.01
2.6	Twitter2013	62.1	60.17	63.09	60.2	60.78
	Twitter2014	60.87	55.45	60.18	56.34	59.35
	Sarcasm	46.51	51.11	**54.71**	44.25	43.11
	SMS	**68.85**	**64.39**	**66.6**	**65.19**	**60.06**
	LiveJournal	**67.16**	**67.42**	**66.74**	**67.01**	**68.43**
	Todas	64.65	62.3	63.86	62.87	**61.24**

que as hipóteses levantadas para as modificações mostraram-se válidas.

A exceção continua sendo a base de Sarcasm, que obteve o resultado mais baixo, mesmo assim somente 15.05 pontos do primeiro colocado do *benchmark* e 120% melhor que o resultado obtido na versão anterior deste trabalho.

Destaque, também, para os resultados da base de LiveJournal, somente 6.41 pontos do primeiro colocado do *benchmark*, mostrando que o modelo é competitivo.

De forma a buscar melhorar ainda mais os resultados, pretende-se, nas próximas versões, incrementar as bases de treinamento dos modelos, bem como melhorar as funções da PG. Além disso, modificações na função de criação de indivíduos serão testadas, de forma a buscar criar indivíduos válidos já na primeira geração, o que provavelmente facilitaria a convergência para um bom resultado.

[Ver se falta discutir mais alguma coisa aqui na conclusão]

REFERENCES

[1] B. Liu, "Sentiment analysis: A multifaceted problem," *IEEE Intelligent Systems*, vol. 25, no. 3, pp. 76–80, 8 2010.

[2] R. Feldman, "Techniques and applications for sentiment analysis," *Commun. ACM*, vol. 56, no. 4, pp. 82–89, Apr. 2013. [Online]. Available: http://doi.acm.org/10.1145/2436256.2436274

[3] N. Guimaraes, L. Torgo, and A. Figueira, "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)*, 2016, pp. 463–471.

[4] S. Vohra and J. Teraiya, "A comparative study of sentiment analysis techniques," *Journal JIKRCE*, vol. 2, no. 2, pp. 313–317, 2013.

[5] A. Bordin-Jr, C. Camilo-Jr, N. Felix, and T. Rosa, "Aplicando programação genética na geração de classificadores de sentimento," *Congresso Brasileiro de Inteligência Computacional (CBIC 2017)*, 2017.

[6] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams Engineering Journal*, vol. 5, no. 4, pp. 1093–1113, 2014.

[7] M. Araújo, P. Gonçalves, and F. Benevenuto, "Métodos para análise de sentimentos no twitter," 2013.

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

[9] N. F. McPhee, R. Poli, and W. B. Langdon, "Field guide to genetic programming," 2008.

[10] A. Patelli, "Genetic programming techniques for nonlinear systems identification," Ph.D. dissertation, "Gh. Asachi" Technical University of Iasi, Romania, 2011.

[11] O. Wijksgatan and L. Furrer, "Gu-mlt-lt: Sentiment analysis of short messages using linguistic features and stochastic gradient descent," *Atlanta, Georgia, USA*, vol. 328, 2013.

[12] P. P. Balage Filho, L. V. Avanço, T. A. S. Pardo, M. d. G. V. Nunes *et al.*, "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.

[13] A. Mudinas, D. Zhang, and M. Levene, "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*. ACM, 2012, p. 5.

[14] A. Pak and P. Paroubek, "Twitter based system: Using twitter for disambiguating sentiment ambiguous adjectives," in *Proceedings of the 5th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, 2010, pp. 436–439.

[15] R. Prabowo and M. Thelwall, "Sentiment analysis: A combined approach," *Journal of Informetrics*, vol. 3, no. 2, pp. 143–157, 2009.

[16] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, 2002, pp. 79–86.

[17] N. Malandrakis, A. Kazemzadeh, A. Potamianos, and S. Narayanan, "Sail: A hybrid approach to sentiment analysis." 2013.

[18] P. D. Turney and M. L. Littman, "Unsupervised learning of semantic orientation from a hundred-billion-word corpus," *arXiv preprint cs/0212012*, 2002.

[19] P. Nakov, A. Ritter, S. Rosenthal, F. Sebastiani, and V. Stoyanov, "Semeval-2016 task 4: Sentiment analysis in twitter." 2016.

[20] F.-A. Fortin, F.-M. De Rainville, M.-A. Gardner, M. Parizeau, and C. Gagné, "DEAP: Evolutionary algorithms made easy," *Journal of Machine Learning Research*, vol. 13, pp. 2171–2175, jul 2012.

[21] S. Luke, G. C. Balan, and L. Panait, "Population implosion in genetic programming," in *Genetic and Evolutionary Computation Conference*. Springer, 2003, pp. 1729–1739.

[22] A. Chehouri, R. Younes, J. Perron, and A. Ilinca, "A constraint-handling technique for genetic algorithms using a violation factor," *arXiv preprint arXiv:1610.00976*, 2016.