

Análise de sentimentos em comentários de filmes

Uma comparação entre modelos de
Processamento de Linguagem Natural

Octávio Marchi Parisi¹ e Thalles Domician Fonseca de Oliveira¹

Abstract—This article explores the data mining using natural language processing models (Naive Bayes, Decision Tree e Max Entropy) in order to identify the text polarity. Comments written in Portuguese for movies. The comments are treated and later classified with the natural language processing models. At the end, the results obtained by the models are compared and the conclusions are presented.

I. INTRODUÇÃO

A análise de sentimentos, também conhecida como mineração de opinião é um campo que consiste em extrair e analisar emoções, opiniões, sentimentos, avaliações sobre produtos, eventos, serviços ou qualquer outro assunto que seja possível se ter opinião. Também pode ser chamada de extração de opinião, mineração de sentimentos, análise de subjetividade, análise de afeições, análise de emoção, revisão de mineração, etc. [6].

Com a explosão das redes sociais e a grande quantidade de opiniões de usuários expressadas na internet as grandes empresas estão muito interessadas na opinião que os usuários compartilham na internet a fim de tomar decisões mais precisas e corretas nos seus negócios com relação aos seus produtos e serviços. Os usuários também tem interesse nas opiniões das outras pessoas na hora de tomar decisões sobre quais serviços e produtos comprar, quais estão com melhores características e com melhor aceitação no mercado [12].

O objetivo deste trabalho é comparar algoritmos que processam linguagem natural e testar técnicas de processamento de texto para analisar sentimentos de comentários em português de filmes, séries e documentários postados em sites especializados em cinema. Os textos foram analisados por cada palavra contida em seu corpo, ou seja, unigramas dentro de cada comentário. Esses modelos foram escolhidos pela simplicidade e facilidade de tratar os dados.

O trabalho de Pang [9] utiliza os modelos Naive Bayes e Máxima Entropia para classificar comentários de filmes, porém com um *corpus* (entende-se por *corpus* o conjunto de dados utilizados na pesquisa) em inglês, apresenta resultados satisfatórios na classificação dos comentários. O trabalho de Nascimento [8] utiliza o modelo Naive Bayes, mas o aplica em *tweets*, a fim de classificar a polaridade desses textos com foco em notícias em português-brasileiro. Além disso existem pesquisas que utilizam outras ferramentas e outras

formas de classificar os sentimentos em textos, além de outras línguas, podem ser encontradas em [10], [3], [12] e [13].

Nesse trabalho foram analisados comentários de filmes feitos por usuários do site *Adoro Cinema*². Os comentários estão em português, feitos ao longo dos anos por usuários com os mais diversos tipos de interesse cinematográfico. Foram coletados comentários dos mais variados tipos, possuindo erros gramaticais, possuem ironias, termos fora de contexto, abreviações, dentre outros.

Durante o processo de desenvolvimento do trabalho houveram muitas tentativas de tratamento do *corpus* da pesquisa, ou seja, muitas formas diferentes de deixar o texto dos comentários o mais normalizado de forma que os modelos de análise de texto pudessem obter os melhores resultados.

Este trabalho está organizado da seguinte forma: na seção II são apresentados os modelos de classificação utilizados no trabalho, o conjunto de dados de utilizados para treinar e testar os algoritmos classificadores, a forma como foram tratados os dados a fim de otimizar a classificação dos modelos, a forma como foram aplicados os dados aos modelos e as ferramentas utilizadas. Na seção III são apresentados os resultados da aplicação ao conjunto de dados de teste após o treinamento dos algoritmos, também é feita a comparação dos resultados de cada algoritmos com métricas utilizadas para avaliar cada aspecto dos resultados. Por fim, na seção IV é apresentada uma conclusão de todo o trabalho, bem como sugestões a trabalhos futuros abrangendo outras áreas de pesquisa e outros conjuntos de dados.

II. METODOLOGIA

Este trabalho busca e recupera os comentários dos filmes através de um *Web Crawler* (rastreador web), uma ferramenta que navega pela página de forma automática e repetitiva copiando e baixando os dados estabelecidos no algoritmo. A execução do rastreador pelas diversas páginas foi rápida devido a inserção de vários processos paralelos que facilitaram a busca e recuperação dos dados.

Foram selecionados três modelos de classificação: Naive Bayes, Árvores de Decisão e Máxima Entropia. Dos dados foram removidos muitos erros de digitação, palavras irrelevantes para o contexto de classificação e palavras que não tem peso positivo ou negativo.

A. Modelos de Classificação

1) *Naive Bayes*: É um algoritmo classificador bastante simples que constrói um conjunto de probabilidades. Essas probabilidades são estimativas geradas pela frequência

¹O. M. Parisi e T. D. F. de Oliveira são estudantes do Bacharelado em Ciência da Computação, Instituto de Ciências Exatas - ICEx, Universidade Federal de Alfenas - UNIFAL-MG, Alfenas-MG, Brasil

²<http://www.adorocinema.com.br>

de cada valor de característica das instâncias de treino e suas posições não são consideradas [8]. Dada uma nova instância, o algoritmo calcula a probabilidade da mesma pertencer a uma classe específica, com base no produto das probabilidades individuais dos valores característicos. Para o cálculo é utilizado o teorema de Bayes e, por essa razão, que esse algoritmo é denominado um classificador de Bayes. A classificação é considerada *naive* (ingênua) porque todos os atributos são independentes da Variável de classe. O algoritmo apresenta, por sua vez, um desempenho satisfatório em diversos cenários de previsão de classes. Estudos experimentais sugerem que este algoritmo tende a aprender mais rapidamente que a maioria dos algoritmos de indução [7].

Dada uma variável de classe Y e um vetor de características dependentes X_1 através X_n , o Teorema de Bayes apresenta a seguinte relação:

$$P(y | x_1, \dots, x_n) = \frac{P(y)P(x_1, \dots, x_n | y)}{P(x_1, \dots, x_n)} \quad (1)$$

Usando a suposição de independência ingênua que

$$P(x_i | y, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = P(x_i | y) \quad (2)$$

Para todo i , essa relação é simplificada para

$$P(y | x_1, \dots, x_n) = \frac{P(y) \prod_{i=1}^n P(x_i | y)}{P(x_1, \dots, x_n)} \quad (3)$$

Uma vez que $P(x_1, \dots, x_n)$ é constante, dada a entrada, podemos usar a seguinte regra de classificação:

$$P(y | x_1, \dots, x_n) \propto P(y) \prod_{i=1}^n P(x_i | y) \quad (4)$$

↓

$$\hat{y} = \arg \max_y P(y) \prod_{i=1}^n P(x_i | y) \quad (5)$$

Podemos usar a estimativa *Maximum a Posteriori* (MAP) para estimar $P(y)$ e $P(x_i | y)$.

O primeiro é então a frequência relativa da classe y no conjunto de treinamento.

2) *Árvores de Decisão*: É uma estrutura hierárquica que traduz uma árvore invertida que se desenvolve da raiz para as folhas. A representação hierárquica mostra uma progressão da análise de dados no sentido de realizar a previsão/classificação. Em cada nível desta árvore tomam-se decisões acerca da estrutura do próximo nível até atingir os nós finais, que são denominadas nós folha [4]. Os algoritmos de árvore de decisão são um dos principais

métodos utilizados de inferência indutiva. Esses algoritmos baseiam-se em métodos de aproximação discreta, onde a função de treinamento é representada por uma árvore de decisão, essas podem ser representadas por um conjunto de regras *if-else*. Para realizar a análise de um parâmetro, a árvore é construída por um processo iterativo, onde um vetor de características é criado. Em cada iteração, a divisão calcula o erro em relação ao dado utilizado, e então, é definido assim o tamanho da árvore. Para a fase de testes, a classificação é feita a partir da raiz, indo até a folha, onde possui o valor da característica aplicada no teste.

Dado vetores de treinamento $X_i \in R^n$, $i = 1, \dots, l$ e um vetor de rótulo $E \in R^l$, uma árvore de decisão recursivamente particiona o espaço de modo que as amostras com os mesmos rótulos sejam agrupadas.

São deixados os dados no nó M ser representados por Q . Para cada divisão de candidatos $\Theta = (j, t_m)$ consistindo em uma característica J e um limite T_m , são particionados os dados $Q_{esq}(\theta)$ e os $Q_{dir}(\theta)$ subconjuntos

$$Q_{esq}(\theta) = (x, y) | x_j \leq t_m \quad (6)$$

$$Q_{dir}(\theta) = Q \setminus Q_{esq}(\theta) \quad (7)$$

A impureza m é calculada usando uma função de impureza H .

$$G(Q, \theta) = \frac{n_{esq}}{N_m} H(Q_{esq}(\theta)) + \frac{n_{dir}}{N_m} H(Q_{dir}(\theta)) \quad (8)$$

São selecionados os parâmetros que minimizam a impureza

$$\theta^* = \operatorname{argmin}_{\theta} G(Q, \theta) \quad (9)$$

É utilizada a recursão nos subconjuntos $Q_{esq}(\theta^*)$ e $Q_{dir}(\theta^*)$ até atingir a profundidade máxima permitida, $N_m < \min_{amostras}$ ou $N_m = 1$.

3) *Máxima Entropia*: Também conhecido por regressão logística binária, ou ainda regressão logística multinomial. É uma técnica que demonstrou ser eficiente em muitas aplicações de processamento de linguagem natural [1]. Ao contrário de Naive Bayes, o modelo de Máxima Entropia não faz suposições sobre as relações entre as características dos dados e, portanto, pode potencialmente funcionar melhor quando os pressupostos de independência condicional não são atendidos [9]. A estimativa de $P(c|d)$ leva o seguinte forma exponencial:

$$P_{ME}(c | d) = \frac{1}{Z(d)} \exp\left(\sum_i \lambda_{i,c} F_{i,c}(d, c)\right) \quad (10)$$

Onde $Z(d)$ é uma função de normalização, $F_{i,c}$ é uma função característica/classe para a função f_i e classe c , definida da seguinte forma:

$$F_{i,c'} = \begin{cases} 1, & n_i(d) > 0 \text{ e } c' = c \\ 0, & \text{caso contrário} \end{cases}$$

Por exemplo, uma característica/função particular de uma classe pode disparar se, e somente se, o bigrama "ainda não gostar de" aparecer e o sentimento do documento tem a hipótese de ser negativo. Os $\lambda_{i,c}$ são parâmetros de peso da característica. A inspeção da definição de P_{ME} mostra que um λ_i grande, c significa que f_i é considerado um indicador forte para a classe c [9].

B. Corpus

Foram encontrados 29.815 filmes e 116.532 comentários para os filmes, com uma média simples de aproximadamente 3,9 comentários para cada filme, vale ressaltar que apenas 8206 filmes tiveram comentários, sendo que aproximadamente 2100 filmes tem apenas um comentário. Os comentários foram organizados em 10 classes de estrelas, onde cada comentário possui uma nota, simbolizada por estrelas, variando de 0,5 a 5,0 estrelas, que representam uma nota, com diferença de 0,5 estrelas a cada classe.

Existem muitos filmes sem comentários e muitos comentários vazios, somando um total de 4064 comentários vazios, ou seja, que não continham texto algum, apenas nota, num total de $\cong 3,49\%$ da base de dados.

Tendo estes dados disponíveis, foi realizada uma análise dos comentários, verificando as palavras que mais se repetem, aquelas que menos se repetem, aquelas que aparecem tanto em comentários positivos quanto negativos.

C. Tratamento dos dados

O objetivo do tratamento dos dados é organizar e sintetizar os comentários coletados e atingir o melhor resultado na aplicação dos modelos de treinamento.

Alguns desafios na análise do *corpus* descritos em [11]

- Textos com erros e sentenças sintaticamente mal formadas que dificultam a busca e classificação dos mesmos;
- Não é trivial distinguir se um comentário é opinião ou fato, e principalmente identificar em um fato se existem opiniões embutidas;
- Os comentários podem conter sarcasmos e ironias, que são difíceis de serem identificados e podem impactar os resultados;
- Um comentário pode referenciar mais de um aspecto de interesse com opiniões diferentes sobre esse aspecto, o que pode confundir a classificação;
- Uso de termos informais da internet, como por exemplo "blz", "fds" e ":", devem ser considerados no vocabulário;

Foi criada uma lista com todos os termos do conjunto de treino, para cada termo foi calculado o número de repetições em comentários positivos e negativos e, ao final, o termo é atribuído como positivo ou negativo de acordo com a maior ocorrência entre as polaridades. De forma mais simples, se um termo aparece 1000 vezes em comentários positivos e 250 vezes em comentários negativos, é atribuído a polaridade positiva em virtude do número superior de vezes que este termo aparece em comentários positivos.

Muitas palavras escritas incorretamente foram encontradas e elas interferem diretamente no treinamento dos algoritmos.

Para amenizar essa situação, resolvemos remover essas palavras, já que não seria possível tratar todas individualmente. Além disso foram removidas as palavras menos relevantes, aquelas que menos aparecem, ou que aparecem em quantidades semelhantes para ambas as polaridades. Os tratamentos de remoção foram:

- Remoção de termos com mais de três caracteres repetidos tais como "gooooostei", "muuuuito", "adoooooorei". Esse filtro foi utilizado pelo fato de que a língua portuguesa não possui palavras que têm mais de três letras iguais repetidas sequencialmente em um mesmo termo, o que implica que os termos que possuem tais características estão incorretas e, de certa forma, podem interferir na classificação pelos modelos preditivos.
- Remoção de termos muito grandes que tenham mais de 20 caracteres, geralmente são erros de digitação, tais como "essefilmeemuito", "poderiatersidomais", "aiudhlfhjksfhlajfa".
- Remoção de termos que aparecem menos de 10 vezes em todos os comentários, visto que, num *corpus* maior que 100 mil comentários, se um termo aparece menos de 10 vezes, esse termo aparece em menos de 0,001% dos comentários.
- Remoção de termos com 2 ou menos caracteres que, no geral, são artigos e termos sem significado.
- Remoção de termos que contém números, geralmente erros de digitação, tais como "247vezes", "1coisa", "30anos" ou apenas números que individualmente não possuem nenhuma polaridade.
- Por último os comentários passaram por um procedimento de remoção de *stopwords*. Para isso, foi utilizado um conjunto de palavras categorizadas como *stopwords* do português brasileiro e, caso alguma palavra fosse encontrada no corpo do comentário analisado, tal palavra seria removida do texto. A lista de *stopwords* utilizadas é a mesma disponibilizada pela ferramenta NLTK, disponibilizadas nas bibliotecas da linguagem Python.

Para tornar o *corpus* mais homogêneo foram separados parte dos dados em dez classes (0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0), de acordo com a avaliação dos usuários, cada classe com a mesma proporção de comentários, aproximadamente 2400 comentários para cada classe e 24000 no total.

Após a separação das classes, para treinar os modelos quanto à polaridade, foram tratados os comentários com classe maior ou igual a 3 como positivos (classificação ≥ 3 = positivo), já os comentários com classificação menor que 3 como negativos (classificação < 3 = negativo). Desta forma foram separados 12.000 comentários positivos e 12.000 comentários negativos.

Novamente foram separados os comentários em dois grupos, o grupo de teste (30%, com 7.200 comentários) e o grupo de treino (70%, com 16.800 comentários). Após todo esse processo foi aplicado o conjunto de treino aos modelos de classificação.

D. Aplicação dos modelos ao conjunto de teste

Tendo feito o tratamento do conjunto de dados, deixando-o mais homogêneo, foi utilizada a implementação padrão da biblioteca NLTK¹ (Natural Language Toolkit) que é um kit de ferramentas que implementa muitos modelos de tratamento, aprendizagem, recursos léxicos para a linguagem de programação Python². O NLTK implementa todos os modelos de classificação utilizados neste trabalho.

III. ANÁLISE E DISCUSSÃO DOS RESULTADOS

Após aplicar os modelos de classificação ao conjunto de treino foi utilizado o conjunto de teste para medir a qualidade e precisão dos modelos treinados.

Para validar os modelos treinados foi gerada a matriz de confusão de cada modelo. A matriz de confusão, também chamada de matriz de classificação, é uma tabela que ajuda a validar os modelos de aprendizado supervisionado. Nela pode-se visualizar as previsões dos modelos e compará-las com os valores reais do conjunto de treino. Os valores da matriz de confusão são utilizados para calcular algumas métricas utilizadas para medir a qualidade dos modelos de aprendizado.

As tabelas a seguir I, II e III mostram as matrizes de confusão dos resultados do modelo Naive Bayes, Árvore de Decisão e Máxima Entropia, respectivamente. A matriz de confusão tem como entrada a quantidade de comentários verdadeiros positivos (V_P), falsos positivos (F_P), falsos negativos (F_N) e verdadeiros negativos (V_N).

TABLE I
MATRIZ DE CONFUSÃO - Naive Bayes

Categoria	Real Positivos	Real Negativos	Total
Predição Positivo	2758	615	3373
Predição Negativo	850	2977	3827
Total	3608	3592	7200

TABLE II
MATRIZ DE CONFUSÃO - ÁRVORE DE DECISÃO

Categoria	Real Positivos	Real Negativos	Total
Predição Positivo	2756	1039	3795
Predição Negativo	852	2553	3405
Total	3608	3592	7200

TABLE III
MATRIZ DE CONFUSÃO - MÁXIMA ENTROPIA

Categoria	Real Positivos	Real Negativos	Total
Predição Positivo	2982	688	3670
Predição Negativo	626	2904	3530
Total	3608	3592	7200

As métricas para avaliação dos modelos são:

¹<http://www.nltk.org/>

²<https://www.python.org/>

- A **Sensibilidade** ou *Recall*, que é a capacidade que o modelo tem de gerar resultados positivos aos comentários que realmente são positivos. Utiliza a fórmula:

$$\text{Sensibilidade} = \frac{\text{Acertos Positivos}}{\text{Total de Positivos}}$$

$$\text{Sensibilidade} = \frac{V_P}{V_P + F_N}$$

- A **Especificidade**, que é a capacidade que o modelo tem de gerar resultados negativos aos comentários que realmente são negativos. Utiliza a fórmula:

$$\text{Especificidade} = \frac{\text{Acertos Negativos}}{\text{Total de Negativos}}$$

$$\text{Especificidade} = \frac{V_N}{V_N + F_P}$$

- A **Acurácia** é a soma total de acertos divididos pelo total de dados. Esta medida pode sofrer interferência por desbalanceamento no conjunto de dados.

$$\text{Acurácia} = \frac{\text{Total de Acertos}}{\text{Total de Dados no Conjunto}}$$

$$\text{Acurácia} = \frac{V_P + V_N}{P + N}$$

- **Eficiência** é a média aritmética entre Sensibilidade e Especificidade.

$$\text{Eficiência} = \frac{\text{Sensibilidade} + \text{Especificidade}}{2}$$

- **Coefficiente de correlação de Matthews**, equivalente ao Coeficiente **PHI**, assume valores entre -1 e +1, onde um valor igual a +1 corresponde a predição perfeita, 0 corresponde a predição completamente aleatória e -1, a predição inversa. E tal coeficiente é dado por:

$$A = (V_P + F_P)$$

$$B = (V_P + F_N)$$

$$C = (V_N + F_P)$$

$$D = (V_N + F_N)$$

$$PHI = \frac{V_P \times V_N + F_P \times F_N}{\sqrt{A \times B \times C \times D}}$$

Quando qualquer uma das quatro operações no denominador for zero, o denominador pode ser arbitrariamente fixado em 1, resultando em um PHI de zero [2].

Tendo as matrizes de confusão I, II e III, foram calculadas as métricas para cada modelo.

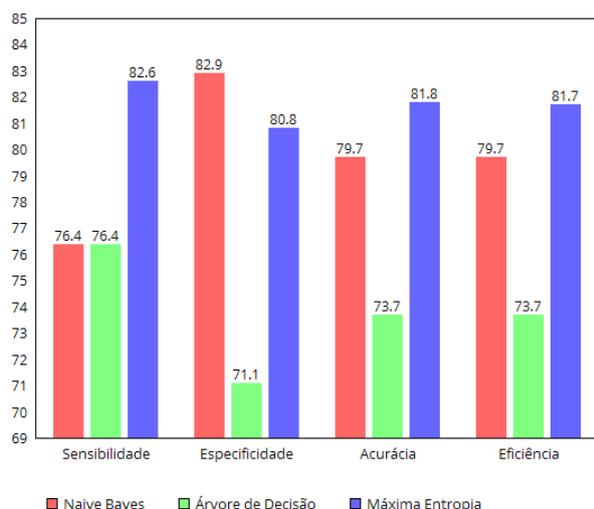
TABLE IV

TABELA COMPARATIVA COM AS MÉTRICA CORRELAÇÃO DE MATTHEWS

Parâmetro	N. Bayes	Árv. de Decisão	Máx. Entropia
Corr. Matthews	0.59	0.48	0.64

Observando o gráfico 1 e a tabela IV podemos concluir que o modelo de Máxima Entropia se destacou com os

Fig. 1. Gráfico comparativo com as Métricas



melhores resultados nas métricas utilizadas neste trabalho. Apesar disso, todos os modelos apresentam resultados muito bons, comparados entre si. Trabalhos sugerem que a capacidade humana de avaliar a subjetividade em um texto varia de 72% [14] e 85% [5]. Sendo assim, os modelos utilizados neste trabalho podem superar as capacidades humanas. O desempenho dos modelos Naive Bayes e Máxima entropia obtiveram os melhores resultados, o que já foi demonstrado em outros trabalhos [8] e [9]. O modelo de Árvores de Decisão obteve bons resultados, porém, não tão bons quanto os demais.

IV. CONCLUSÃO

Neste trabalho foi apresentada uma abordagem para detecção de polaridade de sentimentos em comentários em português de filmes. Foi efetuada a comparação em diferentes métricas para medir a eficácia de cada modelo. A partir da análise dos resultados pode-se concluir que ambos os modelos utilizados neste trabalho possuem características e resultados muito satisfatórios para tratamento de textos em língua portuguesa e, apesar de diferenças nos resultados comparativos, todos tiveram uma boa precisão. Os resultados da classificação de sentimentos em comentários de filmes com os modelos propostos neste trabalho são satisfatórios se comparados à capacidade humana de avaliar a subjetividade em textos. Em termos de resultado relativo pode-se observar que o modelo de *Máxima Entropia* obteve melhores resultados em quase todas as métricas utilizadas, apesar da classificação não ser tão rápida quando o modelo *Naive Bayes*. Os problemas citados em II-C puderam ser observados em grande parte dos comentários, mas ainda assim os modelos conseguiram bons resultados. Portanto, acredita-se que o objetivo da pesquisa foi alcançado.

A utilização de textos em português ainda é escasso em trabalhos científicos de análise de sentimentos, por isso este trabalho pode ser visto como suporte para outras pesquisas voltadas para textos em português. Algumas sugestões ficam,

como o tratamento dos textos utilizando apenas o radical das palavras, ou ainda o tratamento utilizando n-gramas (conjunto com N palavras). Como sugestão de trabalhos futuros, podemos citar a utilização desses modelos em outros contextos de comentários, também em português, visto que existem outros grandes sites com muitos comentários e críticas voltados para livros, músicas, aplicativos de celular, etc. Como *Skoob*³ (Livros), *Google Play*⁴ (aplicativos de celular, músicas, filmes, livros, etc), entre outros.

REFERENCES

- [1] A. L. Berger, V. J. D. Pietra, and S. A. D. Pietra. A maximum entropy approach to natural language processing. *Computational linguistics*, 22(1):39–71, 1996.
- [2] P. H. F. da Silva. Medidas do valor preditivo de modelos de classificação aplicados a dados de crédito. 2008.
- [3] V. de Sales Moreira, S. W. Siqueira, L. Andrade, and M. Pimentel. Análise de sentimentos: Comparando o uso de ferramentas e a análise humana. 2016.
- [4] M. A. dos Santos Rodrigues. Árvores de classificação. *Monografia. Universidade dos Açores. Departamento de Matemática.*, 2005.
- [5] P. Golden. Write here, write now. Article, May 2011.
- [6] B. Liu. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167, 2012.
- [7] A. C. Martins, J. M. Marques, and P. D. Costa. Estudo comparativo de três algoritmos de machine learning na classificação de dados electrocardiográficos. *Trabalho (Mestrado em Informática Médica)–Universidade do Porto, Porto*, 2009.
- [8] P. Nascimento, R. Aguas, D. Lima, X. Kong, B. Osiek, G. Xexéo, and J. Souza. Análise de sentimento de tweets com foco em notícias. In *Brazilian Workshop on Social Network Analysis and Mining*, 2012.
- [9] 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*, pages 79–86. Association for Computational Linguistics, 2002.
- [10] J. C. Reis, P. Gonçalves, M. Araújo, A. C. Pereira, and F. Benevenuto. Uma abordagem multilíngue para análise de sentimentos. *IV Brazilian*, 2015.
- [11] C. A. S. Rodrigues, L. L. Vieira, L. Malagoli, and N. Timmermann. Mineração de opinião/análise de sentimentos. *Trabalho acadêmico, Universidade Federal de Santa Catarina, Florianópolis. www.inf.ufsc.br/~alvares/INE5644/MineracaoOpiniao.pdf*, 2010.
- [12] L. M. Santos, A. A. A. Esmín, A. L. Zambalde, and F. M. Nobre. Twitter, análise de sentimento e desenvolvimento de produtos: Quanto os usuários estão expressando suas opiniões? *Prisma. Com*, (13), 2017.
- [13] V. F. Schmitt. Uma análise comparativa de técnicas de aprendizagem de máquina para prever a popularidade de postagens no facebook. 2013.
- [14] J. Wiebe, T. Wilson, and C. Cardie. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2):165–210, 2006.

³<https://www.skoob.com.br>

⁴<https://play.google.com/store>