Identificação automática de ironia: um caso de estudo no Twitter

Yulli Dias Tavares Alves
yulli.dias@hotmail.com
Centro Federal de Educação Tecnológica de Minas Gerais
Belo Horizonte, Minas Gerais, Brasil

Daniel H. Dalip
hasan@cefetmg.br
Centro Federal de Educação Tecnológica de Minas Gerais
Belo Horizonte, Minas Gerais, Brasil

RESUMO

A análise de sentimentos tem sido aplicada, devido a grande quantidade de informações produzidas nas redes sociais, para investigar a opinião de usuários sobre produtos, marcas e notícias. No entanto, a análise de ironia ainda é um desafio, uma vez que a ironia muda o sentido do texto. Este trabalho tem como objetivo detectar ironia em textos do Twitter, os tuítes. Ao contrário de outros trabalhos que utilizam a língua inglesa ou uma tradução para o português, esse trabalho teve como foco a língua portuguesa. Permitindo assim, uma contribuição para análise de sentimentos nesse idioma. Para isso foi construído um *dataset* a partir da coleta de tuítes irônicos e tuítes a priori não rotulados, que foram manualmente classificados por voluntários. A representação textual inclui a criação de atributos de bag-of-Words (BOW) e bag-of-n-grams. O *dataset* foi usado para construir um modelo de Máquina de Vetores de Suporte (SVM, do original em inglês Support Vector Machine) que foi avaliado pelo método de validação cruzada K-fold. Dentre as representações avaliadas, o melhor resultado foi obtido utilizando-se bag-of-n-grams em que foi obtido 68% para a métrica macro-f1 e 73% para a acurácia. Como resultado deste trabalho tem-se a publicação de um artigo no Webmedia 2019 e como contribuições a disponibilização do código e o *dataset* utilizado.

KEYWORDS

classificação supervisionada, SVM, análise de redes sociais, detecção de ironia

1 INTRODUÇÃO

O intenso uso de redes sociais tem gerado uma grande quantidade de informações sobre os mais diversos temas. Empresas têm aproveitado essas informações para identificar automaticamente a opinião de consumidores em relação à produtos e marcas, visando analisar críticas e sugestões. Além de empresas, pesquisadores exploram o posicionamento nas redes a respeito de temas como política e religião [24]. Para analisar essa grande massa de dados, técnicas de classificação automática são utilizadas para tarefas como análise de sentimento. Essas tarefas visam identificar a opinião dos autores a partir de conteúdo em formato de texto. Uma abordagem comumente utilizada para análise de sentimento é o uso de dicionários léxicos com informações sobre a polaridade das palavras (i.e., se a palavra é positiva ou negativa).

Contudo um dos obstáculos da análise de sentimento é o uso de recursos de linguagem que mudam o sentido das palavras (e.g., ironia). A ironia é um aspecto importante a ser considerado, pois pode inverter o sentido de palavras positivas e negativas. Sendo assim, ao não considerar a existência de ironia em um texto irônico tem-se a interpretação do sentimento ou opinião inversa, o que pode ser desastroso para diversos domínios, como por exemplo análise de satisfação de clientes.

Todavia, a detecção de ironia é uma tarefa difícil, devido a natureza intrínseca de interpretação do ser humano [17]. Em mensagens curtas a identificação de ironia torna-se ainda mais difícil. Como exemplo temos as mensagens enviadas a partir do Twitter, os tuítes, que são limitados a no máximo 280 caracteres. Além disso, comparado à outras redes sociais que permitem textos maiores o Twitter é mais informal, envolvendo uso de gírias e abreviações, o que contribui para uma maior ambiguidade de seus textos [27]. Por isso, a detecção automática de ironia no Twitter não é trivial e requer ferramentas específicas de linguagem [4].

Diante do exposto, neste trabalho é aplicado e avaliado um classificador binário de textos, a fim de detectar o uso de ironia. A metodologia proposta faz uso do classificador automático *Support Vector Machine* (SVM) e os atributos do modelo foram gerados utilizando as representações: bag-of-Words[1](BOW) e bag-of-n-grams. Por fim, foi realizada uma avaliação experimental com o intuito de comparar as abordagens propostas.

Como resultados deste trabalho temos uma publicação [2] no Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia) que é o principal evento sobre Web e Multimídia no Brasil. Como contribuições deste trabalho tem-se a disponibilização do código de desenvolvimento[2], incluindo o código da validação cruzada que utiliza as partições de validação e teste. Essa abordagem ainda não foi implementada pelas principais bibliotecas da área de aprendizado de máquina. Também será disponibilizado a lista de correção de termos da língua portuguesa utilizados neste trabalho e os *datasets*. Além disso, concluiu-se que não existe uma diferença significativa em usar bag-of-words ou bag-of-n-grams para a métrica macro-f1.

O restante deste trabalho está organizado da seguinte maneira. A Seção 2 apresenta alguns conceitos como tuíte, processamento da linguagem natural, representação textual, aprendizado de máquina e teste T. A Seção 3 discute trabalhos relacionados. A Seção 4 detalha o modelo proposto. A Seção 5 apresenta um estudo comparativo através de experimentos, enquanto a Seção 6 conclui o trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Nessa seção serão discutidos alguns dos conceitos utilizados como base para a elaboração deste trabalho. O processo de classificação de ironia em tuítes pode ser divido em etapas sequenciais que são realizadas com o objetivo de classificar um tuíte (Figura 1). A primeira etapa é a coleta dos dados para se obter a base. Tendo os dados disponíveis, é realizado o processamento de linguagem natural nos tuítes com o objetivo de manter um vocabulário mais

[1]Método utilizado no processamento de linguagem natural para representar o texto.
[2]Repositório no Github com o código desenvolvido. https://github.com/yullidias/AutomaticIronyDetection/tree/tcc

padronizado. A partir desse vocabulário, o texto pode ser representado utilizando uma técnica escolhida. Essa representação textual é utilizada como entrada para um método de aprendizado de máquina. Durante os experimentos pode-se construir vários modelos, uma forma de compará-los é por meio dos métodos estatísticos.

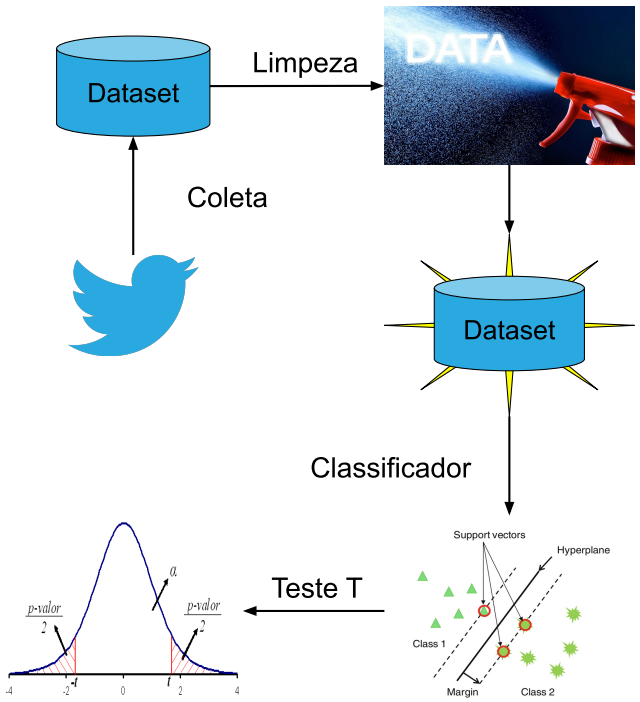

Figura 1: Etapas do trabalho.
Fonte: Adaptado de [14, 21, 31, 36]

2.1 Tuíte

O tuíte (do inglês *tweet*) é uma mensagem publicada no Twitter que pode conter texto, fotos, GIF e/ou vídeo [34]. A Figura 2 exemplifica um tuíte e identifica as possíveis ações que podem ser realizadas a partir dele: responder, retuitar, retuitar com comentário, curtir e compartilhar. O compartilhamento permite enviar o tuíte por mensagem para outro usuário do Twitter, favoritar ou copiar o link. A ação curtir deve ser utilizada para indicar que o usuário gostou do conteúdo [32]. O retuíte, identificado através de um ícone, é uma republicação do tuíte. Essa função permite que o autor ou outros usuários possam compartilhar rapidamente o conteúdo com os seus seguidores, sendo possível adicionar um comentário [33]. Caso não seja adicionado um comentário, o retuíte será um novo tuíte com o mesmo conteúdo do tuíte de original. A ação responder deve ser utilizada quando um usuário deseja responder outro. No tuíte de resposta é indicado a quem o usuário esta respondendo [34](Figura 2).

2.2 Processamento de linguagem natural

Um texto é composto por símbolos que podem ser divididos em dois grupos: os símbolos que estão entre as palavras, chamados

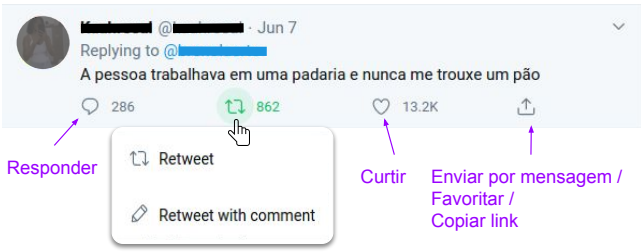

Figura 2: Exemplo de tuíte
Fonte: A autora

separadores, e os símbolos que formam as palavras, chamados de termo. Uma característica importante sobre os símbolos é sua distribuição não uniforme no texto, havendo dessa forma termos que são mais frequentes que outros. Por exemplo, considerando-se somente as letras (a-z) percebe-se que as vogais são mais frequentes que as consoantes em textos [3]. O processamento da linguagem natural (PLN) é um procedimento que realiza transformações em textos que pode ser divido em análise léxica, eliminação de *stopwords* e eliminação de *stemming*.

A análise léxica tem como objetivo principal transformar o texto em um conjunto de termos. Além disso, pode ser feito outras transformações como colocar todos os termos em maiúsculo ou minúsculo e padronizar o vocabulário. Nessa padronização, o termo que casa com um padrão pré-estabelecido pode ser substituído por uma *tag* que o representa. Por exemplo, os dígitos (0-9) podem ser substituídos pela *tag* número. Essa abordagem é útil porque os números são parte de uma grande estrutura, como "10 reais"ou "3º Parágrafo do artigo", ou são ambíguos, como 2000 quilômetros ou 2000 carros. Assim, quando retirados de seu contexto, os números não possuem significados ou são muito vagos [3, 12].

Além dessas transformações, na análise léxica pode ser feito a correção de termos. Essa correção é essencial não só para aprimorar a facilidade de ser ler um texto mas também para obter-se uma alta performance das técnicas de PLN [8]. Os seres humanos possuem um sistema de processamento de linguagem robusto, em que a capacidade de compreensão textual pode facilmente superar erros de digitação, erros de ortografia e a completa omissão de letras ao ler [6]. Essa capacidade humana é amplamente explorada na Internet pela frase abaixo.

> "Nõa imortpa a oderm das ltreas drtneo de uma pvarala, bsata que a pmrireia e a úmtila etjasem no lguar crteo praa que vcoê enednta tduo."

Embora as palavras estejam embaralhadas, é bem provável que a frase acima tenha sido compreendia pelo leitor. Isso porque, conforme a mesma, não importa a ordem das letras que formam uma palavra, desde que a primeira e última letra estejam no lugar certo será possível entender o que está escrito [6]. Essa capacidade humana, entretanto, é um grande desafio enfrentado pela PLN, em que os erros na estrutura de frases impactam fortemente seus objetivos [9]. Além disso, em textos coloquiais, como e-mails, mensagens de texto e blogs, há um maior número de erros. Isso acontece porque muitas vezes, os autores preferem expressões

informais ao invés de expressões formais e corretas. Nesse contexto coloquial a correção de erros se torna ainda mais importante [8].

Assim como as letras, a frequência de termos no texto também possuí como característica uma distribuição não uniforme, sendo a número de termos inversamente proporcional a sua frequência. Embora seja difícil medir formalmente a quantidade de informação presente em um texto, pode-se utilizar dessa distribuição como uma forma de quantificar a informação. Por exemplo, um termo que aparece muitas vezes não contém muita informação. Tipicamente poucos termos ocorrem em 50% do texto, por isso, palavras muitos frequentes, denominadas *stop words*, podem ser retiradas, uma vez que carregam pouco poder discriminativo na linguagem natural. Porém, eliminar as *stopwords* é uma simplificação que reduz o vocabulário mas trás um custo. Por exemplo, caso uma máquina de busca remova as *stopwords* de seu índice, a busca pela banda "the who"irá falhar, por esse motivo as máquinas de busca incluem todos os termos em seu índice [3].

Um outra abordagem utilizada para reduzir o tamanho do vocabulário consiste em remover variações sintáticas como plural, gerúndio, diminutivo, feminino e masculino reduzindo o termo ao seu radical, essa abordagem é chamada de *stemming*. Ao aplicá-la nos termos gatas, gatinhas, gata e gato esses seriam reduzidos ao radical "gat".

Neste trabalho utilizou-se o processamento da linguagem natural para corrigir erros ortográficos, adicionar *tags*, remover caracteres que aparecem mais de duas vezes em sequência, remover acentos e manter o texto em minúsculo. Optou-se, também, por não utilizar técnicas de redução de vocabulário como a eliminação de *stopwords* e o *stemming* de termos nos tuítes.

2.3 Representação textual

Um texto pode ser representado por meio das técnicas de bag-of-Words e bag-of-n-grams. Bag-of-words (BOW) é um modelo em que um texto é representado por um conjunto não ordenado de palavras, sendo as posições ignoradas e mantendo-se apenas a aparição de cada termo [20] que, no presente estudo, foi ponderada pela frequência do termo, ou seja, a quantidade de vezes que o termo aparece no texto. BOW é um modelo simples e muito utilizado para extração de atributos de textos. Bag-of-n-grams, por sua vez, consiste em sequências de palavras de tamanho n [16]. Um bigrama, por exemplo, é uma sequência de duas palavras. Quando se gera um n-gram a partir de um documento, é gerado um conjunto dos termos combinados sequencialmente, levando em consideração uma janela de n termos. A Figura 3 exemplifica a representação de BOW (ou unigrama), bigrama e trigrama para a frase "Hoje é o dia que o almoço sai no jantar". Nessas representações o termo é composto, respectivamente, por uma, duas e três palavras.

2.4 Aprendizado de máquina

Aprendizado de máquina (do inglês *Machine Learning*) é uma área da inteligência artificial cujo o objetivo é extrair automaticamente informações de dados por meio de métodos computacionais e estatísticos [21]. Um método de aprendizado de máquina recebe como entrada os dados (atributos) e as respostas (rótulos) para esse dados e produz como saída as regras. Essas regras podem ser

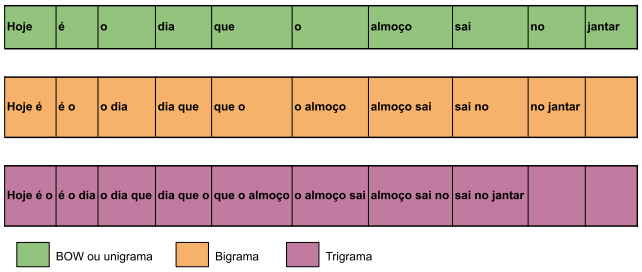

Figura 3: Exemplo bag-of-words e bag-of-n-gram.
Fonte: A autora

aplicadas em novos dados, para os quais não se sabe o rótulo, para produzir as respostas. [10].

A máquina de vetores de suporte (SVM, do original em inglês *Support Vector Machine*), desenvolvida por Corinna Cortes e Vladimir Vapnik [11], é um método de aprendizado de máquina para classificação binária. Seu objetivo é aprender o melhor hiperplano de separação dos dados de treinamento. Assim, para classificar novos dados, é necessário apenas verificar em qual lado do hiperplano eles estão [10].

Para encontrar o hiperplano de separação, o SVM realiza dois passos. No primeiro passo, os dados de entrada são mapeados usando uma função de transformação escolhida a priori, chamada de *kernel*, para uma nova representação de alta dimensão em que seja possível definir um hiperplano de separação [21]. No segundo passo, o hiperplano ótimo é computado tentando maximizar a distância entre o hiperplano e os pontos mais próximos de cada classe, ou seja, tentando maximizar a margem. Isso permite que o hiperplano generalize bem novas amostras [10].

Os vetores de suporte são os dados de treino utilizados para construir o hiperplano de separação [11, 21]. Um exemplo de vetores de suporte para a classificação de duas classes distintas, estrelas e triângulos, é representado na Figura 4, em que é possível verificar que os três vetores de suporte, circulados na imagem, contêm a informação necessária para definir o hiperplano e separar as duas classes. É possível perceber, também, que os vetores de suporte representam os dados mais difíceis de serem classificados.

Além disso, a capacidade de generalização do melhor hiperplano está diretamente relacionado ao número de vetores de suporte, isto é, o hiperplano com menor complexidade possui a maior margem [21]. Esse é um motivo pelo o qual o SVM não precisa de muitos dados para conseguir generalizar. Outro importante benefício do SVM é a sua sólida base teórica que está mais próxima da realidade das aplicações práticas. Por isso, o SVM possui diversas aplicações como classificação de letras escritas a mão, classificação de texto e reconhecimento de faces [21].

2.5 Métricas

Algumas métricas são utilizadas para avaliar o desempenho de um modelo de classificação. Neste trabalho será discutido o desempenho com base nas métricas de classe: precisão, revocação e f1-score e métricas de desempenho geral: macro-f1 e acurácia. A seguir é apresentado as fórmulas dessas métricas, em que vp e fp são o número de verdadeiros positivos e negativos,

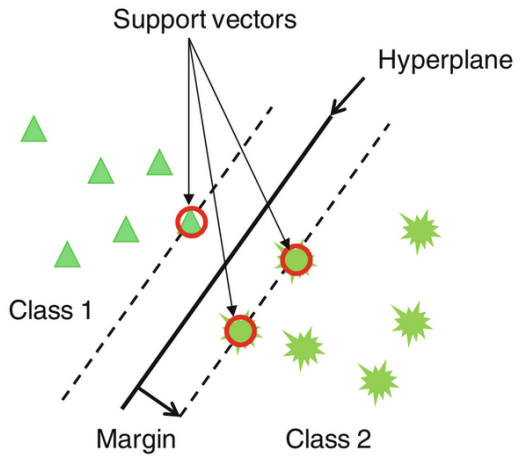

Figura 4: Hiperplano com a maior margem de separação de duas classes.

Fonte: [21]

respectivamente. E fp e fn são o número de falsos positivos e negativos, respectivamente. A precisão diz respeito a quantidade de classificações corretas que o modelo fez dentre todas as suas classificações para a classe positiva. A revocação é a quantidade de classificações corretas que o modelo fez dentre todas as situações de classe positiva. O *f1-score* é a média harmônica entre a precisão e a revocação. A acurácia indica a performance geral do modelo e o macro-f1 é a média do *f1-score* de todas as classes do modelo.

$$\text{precisão} = \frac{vp}{vp + fp} \tag{1}$$

$$\text{revocação} = \frac{vp}{vp + fn} \tag{2}$$

$$\text{f1-score} = \frac{2 \cdot \text{precisão} \cdot \text{revocação}}{\text{precisão} + \text{revocação}} \tag{3}$$

$$\text{macro-f1} = \frac{1}{\text{total de classes}} \sum_i \text{f1-score}_i \tag{4}$$

$$\text{acurácia} = \frac{vp + vn}{vp + vn + fp + fn} \tag{5}$$

2.6 Teste T

Muitos problemas em pesquisa experimental consistem em aceitar ou rejeitar uma afirmação acerca de um parâmetro. Essa afirmação é chamada de hipótese e a decisão sobre a hipótese é chamada de teste de hipóteses, sendo importante destacar que as hipóteses são sempre afirmações sobre a população e não afirmações sobre a amostra. Os procedimentos de teste de hipóteses se apoiam no uso de informações de uma amostra aleatória proveniente de uma população de interesse. Se essa informação for consistente com a hipótese, essa não será rejeitada. Entretanto, se essa informação for inconsistente com a hipótese, concluirá-se que a hipótese é falsa. Se toda a população for examinada, a verdade ou falsidade de uma hipótese não apresentará erros, como isso é geralmente impossível em muitas aplicações práticas, um procedimento de teste de hipóteses deve ser desenvolvido tendo-se em mente a probabilidade de cometer o erros. A probabilidade de cometer o erro de rejeitar a hipótese nula quando ela é verdadeira é denotada por α [25].

Um procedimento prático para aplicação de teste de hipóteses pode ser divido em três etapas: especificar a estatística de teste a ser usada, especificar a localização da região crítica (bilateral, unilateral superior ou unilateral inferior) e especificar os critérios de rejeição (tipicamente o valor de α ou valor-p no qual a rejeição deveria ocorrer) [25].

Em testes de hipóteses para a média de uma população com distribuição normal e variância conhecida, considerando uma amostra aleatória X_1, X_2, ..., X_n pode-se afirmar que a média amostral $\overline{X}$ é um estimador não-tendencioso com média μ_0 e desvio padrão $\sigma/\sqrt{n}$. Nesse caso podemos utilizar a estatística de teste Z_0 (Equação 6) [25].

$$Z_0 = \frac{\overline{X} - \mu_0}{\sigma/\sqrt{n}} \tag{6}$$

Em muitas aplicações práticas a variância (σ^2) é desconhecida, nessas situações o desvio-padrão S da amostra poderá substituir σ nos procedimentos de teste, tendo pouco efeito. Nos casos em que o tamanho da amostra n é pequeno, $n < 40$, pode-se utilizar o teste T. Para realizar esse teste, é necessário que as amostras aleatórias $X_1, X_2, ..., X_n$ sejam provenientes de uma distribuição normal com média μ e desvio-padrão S. Nesse caso, utiliza-se a estatística de teste T_0 (Equação 7). Se a hipótese a ser testada (hipótese nula) for verdadeira, então T_0 (Equação 7) terá uma distribuição t com $n - 1$ graus de liberdade [25]. Para testar se uma amostra possui uma distribuição normal, pode-se usar o teste de Shapiro-Wilk [29].

$$T_0 = \frac{\overline{X} - \mu_0}{S/\sqrt{n}} \tag{7}$$

Uma maneira de reportar os resultados de um teste de hipóteses é estabelecer que a hipótese nula foi ou não rejeitada com um valor especificado de α, ou nível de significância. Essa abordagem, entretanto, impõe a escolha de um nível predefinido de significância e não permite identificar se o valor estatístico calculado estava apenas nas proximidades da região crítica ou se estava muito longe dessa região. Com o objetivo de evitar essas dificuldades, o valor-p tem sido adotado na prática. O valor-p é o menor nível de significância que conduz a rejeição da hipótese nula H_0, ou seja, é o menor valor de α em que a estatística de teste é significativa [25] (Figura 5).

3 TRABALHOS RELACIONADOS

Nesta seção serão apresentados os trabalhos que tratam da identificação de ironia e sarcasmo em textos. Reyes et al [28] propuseram dois modelos de classificação, Árvore de decisão e Naive Bayes, para a detecção automática de ironia. Para a construção do *dataset* foram coletados tuítes irônicos utilizando *#irony* como termo de busca e os tuítes não irônicos foram coletados a partir dos termos *#education*, *#humor* e *#politics*. Os quatro atributos utilizados foram construídos manualmente e consideram as propriedades textuais de textos irônicos, são eles:

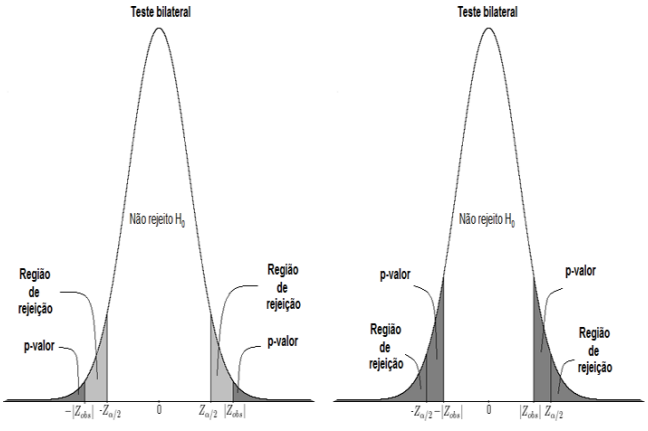

Figura 5: Considerando-se α como a região de rejeição, pode-se observar que na figura da esquerda valor-p < α. Nesse caso, deve-se rejeitar a hipótese nula. Na figura da direita valor-p > a, por isso a hipótese nula não é rejeitada.

Fonte: [1]

assinatura, inesperabilidade, estilo e cenários emocionais. O melhor resultado foi obtido utilizando-se os dados balanceados e a Árvore de decisão com validação cruzada de 10 *folds*. Os resultados foram: acurácia de 72.30%, precisão de 73.60%, revocação de 69.50% e f1-score de 71.50%.

Barbieri e Saggion [4] utilizam o *dataset* produzido por Reyes et al [28]. Como atributos o autor propõe frequência (*gap* entre palavras raras e comuns), linguagem formal e informal, intensidade, sinônimos e ambiguidade. Os classificadores Naive Bayes, Random forest e Árvore de decisão foram utilizados com a validação cruzada com 10 *folds*. Os autores apresentaram um f1-score superior ao encontrado pelo trabalho que criou o *dataset*.

Um dos problemas na identificação de ironia é a divergência entre pesquisadores na definição formal e estrutura de ironia ou sarcasmo [15], pois não é claro se essas figuras de linguagem são essencialmente a mesma coisa ou possuem diferenças superficiais ou significativas. O sarcasmo pode representar um tipo de ironia abertamente agressivo e com um alvo bem definido [5]. Por isso alguns autores consideram sarcasmo e ironia como o mesmo fenômeno [23]. Essa abordagem será utilizada neste trabalho, em que o sarcasmo não será diferenciado da ironia.

Em relação a trabalhos que abordam o sarcasmo, Filatova [15] propõe um classificador de sarcasmo em resenhas de produtos da Amazon, fazendo uma análise qualitativa e quantitativa do conteúdo. Davidov et al. [13] usa um algoritmo semi-supervisionado e obtém bons resultados para classificação de sarcasmo em resenhas da Amazon e textos do Twitter, ambos em inglês. Ling e Klinger [22] propõem um modelo que separa ironia de sarcasmo com 79% de acurácia em uma base balanceada.

A importância da detecção de sarcasmo na análise de sentimentos tem sido discutida por diversos autores. Bouazizi e Ohtsuki [7] fazem uma comparação entre considerar ou não o sarcasmo na análise de sentimentos, usando três métodos de aprendizado de máquina distintos. A base utilizada é um conjunto de tuítes classificados em negativos e positivos. Esse tuítes foram representados em atributos classificados em quatro categorias, atributos relacionados ao sentimento, a pontuação, a sintaxe e ao grau de similaridade do tuíte com um conjunto de 18.959 tuítes que contém a *hashtag* #sarcasmo. Esse trabalho demonstra que a detecção de sarcasmo é um aspecto importante na análise de sentimentos. Todos os métodos de aprendizagem alcançam um resultado superior quando foram extraídos os atributos dos tuítes e o SVM foi o que apresentou o melhor resultado, com uma revocação de 92% e uma acurácia de 89%.

Karthik e Anandhakumar [30] identificam o tipo de sarcasmo, classificando-o em quatro tipos: educado, rude, furioso e sem expressão. Para obter os tuítes os autores utilizaram como termo de busca na API do Twitter as *hashtags #sarcasm*, *#sarcastic*, *#Sarcasm* e *#notSarcasm*. No preprocessamento, foi removido as hashtags e URLs, além de realizar a lematização e stemização. Foram criadas 20 features e utilizou-se o sistema *fuzzy* para obter um f1-score para os tipos de sarcasmo educado, rude, furioso e sem expressão de 82%, 43%, 95% e 89%, respectivamente.

Neste trabalho será realizada identificação de ironia por meio de classificação binária, sendo utilizada para a criação de atributos BOW e bag-of-n-grams e como método de aprendizado o SVM. A base utilizada foi coletada usando a API do Twitter e contempla somente textos em português.

4 METODOLOGIA

Este trabalho propõe uma abordagem para a detecção automática de ironia em tuítes. Para tanto foram coletados tuítes irônicos e não irônicos. Após a coleta e rotulação obteve-se um conjunto de treino representado da seguinte forma $(id_1, t_1, i_1), ..., (id_n, t_n, i_n)$ em que t_j é um tuíte, id_j é um identificador inteiro único e i_j é a classe que indica quando um tuíte é irônico ou não ($i_j = 1$, quando possui ironia e $i_j = 0$, caso contrário).

4.1 Coleta de Dados

A coleta de dados foi feita por meio da API do Twitter[3] e da biblioteca Tweetpy [4], sendo buscados apenas tuítes em português. Com o objetivo de coletar os tuítes irônicos foi utilizado como consulta as *hashtags* #sqn, #soquenao e #ironia. Isso porquê ao utilizar essas *hashtags*, o próprio usuário já definiu o conteúdo como irônico. Com essa abordagem foram obtidos um total de 2.701 tuítes, rotulados como irônicos. Além desses, outros tuítes foram obtidos utilizando como termo de busca as *stopwords*[5], a fim de buscar um número grande de tuítes. Como as *stopwords* são palavras comuns e amplamente utilizadas em textos, buscar tuítes por essas palavras resulta em um grande número de textos retornados. Os tuítes dessa segunda abordagem são a priori não rotulados, ou seja, não se sabe se são classificados como irônicos ou não irônicos. Com essa abordagem foi obtido um total de 517.417 tuítes. Deste total foram selecionados aleatoriamente 5.000 tuítes para que o conjunto total de dados estivesse balanceado[6]

[3] https://developer.twitter.com/

[4] https://www.tweepy.org/

[5] As stopwords utilizadas foram selecionadas a partir das *stopwords* disponibilizadas pelo NLTK e encontram-se em https://github.com/yullidias/AutomaticIronyDetection/blob/tcc/data/stopwordsSelectedForCollectUnlabeledTweets.txt

[6] O número de tuítes selecionados é um pouco maior que o dobro de tuítes irônicos pois dentre eles poderiam haver tuítes irônicos e tuítes que o usuário não saberia classificar. Dessa forma, para que houvesse um balanceamento foi escolhido esse número de tuítes

com os irônicos e 100.000 tuítes, também escolhidos aleatoriamente, para compor a base de pré-processamento.

Foi necessário rotular manualmente o conjunto de 5.000 tuítes. Para isso, dividiu-se os tuítes em 25 planilhas de 200 mensagens estruturadas conforme a Tabela 1. A coluna "N"identifica o número do tuíte na planilha, a coluna "rótulo"foi enviada vazia, pois nela os voluntários deveriam classificar o "tuíte a ser avaliado"como **irônico**, **não irônico** ou **não sei**. Como descrito na Subseção 2.1, um tuíte A pode responder ou retuitar outro tuíte B, dessa forma, B é o contexto de A. Com o objetivo de trazer um contexto para ajudar na classificação, utilizou-se a coluna "Tuíte pergunta". Para um tuíte n_i, se houver um tuíte p_i na coluna "Tuíte pergunta", então o tuíte n_i respondeu ou retuitou com comentário o tuíte p_i, um exemplo disso é o tuíte 1 da Tabela 1. Entretanto, se p_i for vazio, então o tuíte n_i não possui contexto ou é um retuíte sem comentário.

N	Tuíte "Pergunta"	Tuíte a ser avaliado	Rótulo
1	tuíte p_1	tuíte n_1	
2		tuíte n_2	
...	...	...	
200		tuíte n_{200}	

Tabela 1: Exemplo de planilha enviada para os voluntários rotularem manualmente

A Tabela 2 apresenta o resultado do processo de rotulagem por voluntários com o número de tuítes por rótulo. Após a rotulagem, os tuítes n_i que eram um retuíte sem comentário foram removidos, pois embora fossem tuítes diferentes, apresentavam o mesmo conteúdo. Esse resultado é representado na última coluna da Tabela 2. Dessa forma, ao fim desse processo, tivemos como resultado 2406 tuítes rotulados[7].

Rótulo	Número de tuítes	Número de tuítes sem retuítes
Irônico	940	518
Não irônico	3495	1594
Não sei	565	294
Total	5000	2406

Tabela 2: Resultado da rotulagem do conjunto rotulado manualmente

4.2 Processamento de linguagem natural

O pré-processamento foi utilizado padronizar o vocabulário, para isso utilizou-se a base de pré-processamento. Dentre essas transformações estão colocar os termos em minúsculo, a remoção de *hashtags* #sqn, #soquenao e #ironia, remoção de caracteres especiais e caracteres ou pares de caracteres que aparecem mais que duas vezes em sequência. Foi realizado, também, uma substituição de entidades[8] como links, *emojis* e marcações com o objetivo de mapear os atributos equivalentes. Na Tabela 3 há um detalhamento das substituições feitas.

Entidade	Tag substituta
emojis de coração	CORAÇÃO
emojis de raiva	RAIVA
emojis caveira	MORTE
emojis tristes	TRISTE
emojis felizes	FELIZ
emoji risos, haha, kk, rsrs, kaka, hehe	RISOS
data e/ou hora	DATA
@ seguido de a-z, A-Z, 0-9	MARCAÇÃO
# seguido de a-z, A-Z, 0-9	HASHTAG
expressão regular que reconhece urls	URL
expressão regular que reconhece email	EMAIL
números ordinais, reais, telefone, dinheiro	NUMERO
’.’ que aparece duas ou mais vezes	RETICÊNCIAS
’+’	MAIS

Tabela 3: Substituição de entidades

Além das substituições de entidade, foi feito uma correção de termos considerando erros comuns[9], erros de ortografia[10] e erros identificados experimentalmente durante a análise do vocabulário da base de pré-processamento. Esses erros foram consolidados em um dicionário[11] em que a chave era o termo a ser corrigido e o valor era a correção desse termo, esse dicionário contém 1.187 termos corrigidos.

Para a geração dos atributos utilizou-se as técnicas bag-of-Words (BOW) e bag-of-n-grams, sendo utilizados unigrama, bigrama e trigrama. Utilizou-se essa sequência de n-grams pois elas se mostraram boas em melhorar o resultado de classificação de artigos enquanto sequências maiores não apresentaram um aumento expressivo [16]. Essas representações resultaram em, respectivamente, 62.792 e 1.994.188 atributos, gerados a partir da base de pré-processamento. Essa base foi utilizada para definir o vocabulário a ser utilizado nas representações e para a validação do processamento da linguagem natural. Dessa forma, os atributos gerados a partir dela formam o vocabulário no qual os tuítes dos conjuntos de dados serão representados.

4.3 Conjunto de dados

A base completa, formada pelos tuítes irônicos e não irônicos foi utilizada para a avaliação do modelo. Com o objetivo de fazer experimentações, essa foi dividida em subconjuntos. A Tabela 4 exibe o composição desses subconjuntos da base. O subconjunto ironia consiste nos tuítes coletados utilizando como consulta a *hashtag* #ironia e os tuítes rotulados como não irônicos. O mesmo ocorre para os subconjuntos soquenao e sqn, que foram coletados a partir das *hashtags* #soquenao e #sqn, respectivamente. O subconjunto manual contém os tuítes irônicos rotulados manualmente pelos voluntários e os tuítes não irônicos. O

[7] Os tuítes do conjunto rotulado manualmente se encontram em: https://github.com/yullidias/AutomaticIronyDetection/blob/tcc/data/manually_labeled.xlsx

[8] Disponível em https://github.com/yullidias/AutomaticIronyDetection/blob/tcc/data/substituicaoDeEntidades.md, no qual é possível visualizar os emojis utilizados

[9] https://pt.wikipedia.org/wiki/Wikipédia:Lista_de_erros_comuns/Máquinas

[10] https://www.dicio.com.br/erros-de-ortografia/

[11] Dicionário com os termos corrigidos https://github.com/yullidias/AutomaticIronyDetection/blob/tcc/src/error_map.py

subconjunto *hashtags* contém os tuítes irônicos coletados pelas *hashtags* #ironia, #sqn e #soquenao e os tuítes não irônicos. O conjunto base completa, como o nome sugere, contém todos os tuítes independente do método de obtenção ou rótulo.

Conjuntos	irônicos	não irônicos	Total
ironia	496	1594	2090
soquenao	279	1594	1873
sqn	1926	1594	3520
hashtags	2701	1594	4295
manual	518	1594	2112
Base completa	3219	1594	4813

Tabela 4: Número de tweets para cada subconjunto da base

4.4 Avaliação

Na avaliação do classificador foi utilizado o método de validação cruzada K-fold utilizando conjuntos de treino, validação e teste. A Figura 6 exemplifica como os dados são representados nesse método para 5 *folds*. Assim, é possível perceber que para cada partição temos 3 *folds* para treino, 1 para a *fold* validação e 1 *fold* para teste. O classificador é executado para cada partição utilizando o *fold* de validação para variar os parâmetros do classificador e o teste para aplicar os melhores parâmetros encontrados. O código que implementa o validação cruzada K-fold exemplificado na Figura 6 para qualquer K é uma das contribuições deste trabalho[12].

A escolha do valor K deste método envolve um *trade-off* entre viés e variância, sendo os valores k = 5 ou k = 10 tipicamente usados por se mostrarem, empiricamente, resultantes em um equilíbrio entre viés e variância [19]. Neste trabalho foi utilizado o kernel linear e o valor K = 10, sendo 8 *folds* para treino, 1 para validação e 1 para teste. Na validação variou-se o parâmetro C do SVM de 2^{-5} a 2^{15} aumentando o expoente de 2 em 2 como sugerido por Hsu et al. [18]. O parâmetro C escolhido para ser usado no teste, de cada partição, foi aquele que obteve o maior macro-f1.

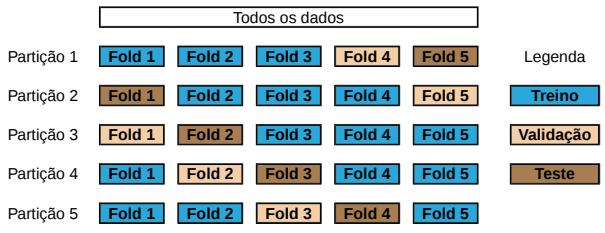

Figura 6: *K-fold cross validation.*
Fonte: A autora

[12]Código genérico para a validação cruzada K-fold está disponível em: https://github.com/yullidias/AutomaticIronyDetection/blob/tcc/src/my_cross_validation.py

4.5 Teste T

O teste T assume que os dados possuem uma distribuição normal. Antes de aplicá-lo, será demonstrado que os dados possuem essa distribuição. Para isso, será utilizado o teste de Shapiro-Wilk. Esse teste é realizado utilizando uma hipótese que assume a normalidade dos dados (Hipótese H_0 and H_1) e estabelece o nível de significância do teste α, normalmente 0.05. Como resultado do teste, obtêm-se a estatística de teste W e o valor-p. Valores pequenos de W, que varia de 0 a 1, indicam que a distribuição não é normal e $W = 1$ indica que a distribuição é normal [29]. Além disso, se o valor-p for maior que α, assume-se que a distribuição é normal e se o valor-p for menor que α, então a distribuição não é normal. Para aplicar o teste de Shapiro-Wilk considerou-se o macro-f1 de cada fold, para as representações BOW e bag-of-n-grams.

Hipótese H_0: A amostra deriva de uma população normal

Hipótese H_1: A amostra não deriva de uma população normal

Para aplicar o Teste t construiu-se uma hipótese acerca das médias do macro-f1 de ambas as representações (Hipótese H_0 and H_1). Em que μ_1 o macro-f1 médio para a representação bag-of-n-grams e μ_2 o macro-f1 médio para a representação BOW. Os resultados para o teste de Shapiro-Wilk e Teste t foram obtidos utilizando-se a biblioteca Scipy [35].

Hipótese H_0: $\mu_1 = \mu_2$

Hipótese H_1: $\mu_1 != \mu_2$

5 EXPERIMENTOS

O método de Aprendizado de Máquina escolhido foi o SVM, uma vez que é um dos métodos com melhor eficácia para classificação automática de texto [37]. Nos experimentos, foi utilizada a implementação do SVM disponível na biblioteca Scikit-learn [26] e o kernel linear por ser altamente recomendado quando o número de atributos é grande e mais expressivo que o número de observações [18]. Para avaliação do classificador foi utilizado o método de validação cruzada *K-fold*. O SVM foi executado para as representações BOW e bag-of-n-grams e utilizou-se o teste T para avaliar se há uma diferença significativa entre o macro-f1 médio das duas abordagens. Além do macro-f1, analisou-se as métricas de f1-score, precisão e revocação como o objetivo de detalhar os resultados obtidos. Além disso, avaliou-se a precisão do modelo para os subconjuntos da base discutidos na Subseção 4.3.

A Figura 7 mostra os resultados para a métrica macro-f1 para as duas representações para cada partição. A partir dela percebe-se que os resultados obtidos não estão dispersos da média para as ambas as representações, o que pode ser confirmado pelo desvio padrão de 0.02 e 0.04 para as representações BOW e bag-of-n-grams, respectivamente (Tabela 5). Cada partição apresentada na Figura 7 foi considerada como uma amostra na realização dos testes de Shapiro-Wilk e teste T.

Aplicando-se o teste de Shapiro-Wilk para verificação de normalidade do macro-f1 para ambas as representações, conclui-se que ambas possuem distribuição normal com 5% de significância (valor-p de 0,5023 e 0,9079 para as representações BOW e bag-of-n-grams, respectivamente). Sabendo-se que as amostras

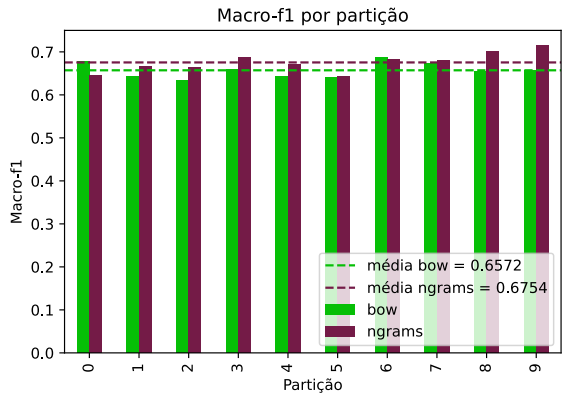

Figura 7: Macro-f1 por partição para as representações BOW e bag-of-n-grams.

Fonte: A autora

	BOW	bag-of-n-grams
macro-f1	0.66(±0.02)	0.68(±0.04)
acurácia	0.70(±0.02)	0.73(±0.04)

Tabela 5: Resultado para as representações BOW e bag-of-n-grams

apresentam uma distribuição normal, foi realizado o teste t (Hipótese H_0 and H_1) para verificar a igualdade entre os macro-f1 médios de cada representação. Com 2% de significância, conclui-se que os macro-f1 médios de BOW e bag-of-n-grams não possuem diferença significativa (valor-p = 0,0590).

A Tabela 6 mostra os resultados de f1-score dos tuítes irônicos e não irônicos, para o BOW e a Tabela 7 para bag-of-n-grams. Em ambas as tabelas, é possível perceber que o modelo teve resultados superiores a 76% para identificar os irônicos e resultados superiores a 50% para identificar os não irônicos. Pode-se concluir que o modelo possui dificuldade de identificar os tuítes não irônicos. A diferença para o f1-score entre as representações foi de 3,45% para os irônicos. Além disso, observando-se a precisão e a revocação dos irônicos nota-se que o modelo tem uma precisão maior que a revocação para a representação bag-of-n-grams, isso significa que quando o modelo diz que o tuíte é irônico ele acerta, mas há irônicos que ele consegue identificar.

Classe	F1-score	Precisão	Revocação
irônicos	0.7708	0.7673	0.7748
não irônicos	0.5435	0.5498	0.5388

Tabela 6: Resultados para a representação BOW

Com o objetivo de identificar se existe diferença para o modelo prever ironia para cada conjunto definido na Subseção 4.3. Foi realizado uma predição para cada um dos subconjuntos para todas as partições. A Tabela 8 exibe os resultados de macro-f1 para cada

Classe	F1-score	Precisão	Revocação
irônicos	0.8053	0.8433	0.7714
não irônicos	0.5455	0.4952	0.6111

Tabela 7: Resultados para a representação bag-of-n-grams

conjunto de dados. Os resultados para a representação bag-of-n-grams, para os tuítes selecionados foram melhores do que a representação BOW. A partir dessa tabela também é possível perceber que os piores resultados foram obtidos para os conjuntos soquenao e manual. O conjunto soquenao é o que possui o menor número de tuítes (Tabela 4) e o manual foram classificados pelas pessoas, essas características podem ser o motivo para a diferença desses resultados em relação aos demais conjuntos.

Conjunto	BOW	bag-of-n-grams
ironia	0.80	0.87
soquenao	0.76	0.83
sqn	0.87	0.92
hashtags	0.87	0.93
manual	0.67	0.81

Tabela 8: Resultados de macro-f1 para as representações BOW e bag-of-n-grams dos cojuntos

A Tabela 9 e Tabela 10 exibem as métricas f1-score, precisão e revocação para os tuítes irônicos e não irônicos. A Tabela 9 exibe os resultados para a representação BOW enquanto a Tabela 10 exibe os resultados para a representação bag-of-n-grams. Considerando a revocação, o conjunto de dados com mensagens rotuladas por voluntários apresentou o pior resultado. Uma hipótese para explicar isso é que a ironia implícita pode ser mais difícil de predizer. Outra hipótese seria que os voluntários responsáveis pela rotulagem hesitaram em classificar tuítes como irônicos, quando não estavam certos da ironia, resultando em um número pequeno de mensagens rotuladas manualmente como irônicas. Caso a última hipótese esteja correta, isso indica que são necessários mais dados desse tipo para se obter melhores resultados.

6 CONCLUSÃO

Neste trabalho foi apresentado e avaliado abordagens baseadas em aprendizado de máquina para detectar a uso de ironia em textos curtos. O texto foi representado, após um pré-processamento, pelas abordagens BOW e n-grams. O resultado encontrado utilizando-se bag-of-n-grams foi 68% para a métrica macro-f1 e 73% para a acurácia. Apesar disso, a diferença para o resultado da representação BOW não é significativa, de acordo com os resultados obtidos no teste T. Ambas representações alcançaram o resultado de f1-score superior a 77% para os irônicos e 54% para os não irônicos.

Assim, é possível perceber que o modelo tem uma dificuldade em predizer os não irônicos. Além disso, observou-se que o conjunto de tuítes manualmente rotulado por voluntários possui o pior resultado em relação aos demais conjuntos. Trabalhos futuros

Classe	Conjunto	*F1-score*	Precisão	Revocação
irônicos	ironia	0.71	0.60	0.88
	soquenao	0.63	0.47	0.93
	sqn	0.89	0.86	0.93
	hashtags	0.91	0.89	0.92
	manual	0.51	0.48	0.53
não irônicos	ironia	0.88	0.95	0.82
	soquenao	0.89	0.99	0.82
	sqn	0.86	0.90	0.82
	hashtags	0.83	0.85	0.82
	manual	0.83	0.84	0.82

Tabela 9: Resultados de predição para a representação BOW

Classe	Conjunto	*F1-score*	Precisão	Revocação
irônicos	ironia	0.81	0.70	0.96
	soquenao	0.72	0.57	0.98
	sqn	0.93	0.90	0.97
	hashtags	0.95	0.93	0.97
	manual	0.72	0.66	0.78
não irônicos	ironia	0.92	0.99	0.87
	soquenao	0.93	1.00	0.87
	sqn	0.91	0.96	0.87
	hashtags	0.91	0.95	0.87
	manual	0.90	0.93	0.87

Tabela 10: Resultados de predição para a bag-of-n-grams

podem aumentar esse conjunto de tuítes. Além disso, pode-se realizar uma abordagem de forma que um tuíte seja rotulado por mais de um voluntário. Dessa maneira, será possível medir a concordância ou discordância entre os voluntários sobre a existência ou não de ironia.

Os resultados desse trabalho foram publicados no WebMedia. Como contribuições tem-se a conclusão que não existe uma diferença significativa em usar bag-of-words ou bag-of-n-grams para a métrica macro-f1, a disponibilização do código de desenvolvimento[13]. Também será disponibilizado a lista de correção de termos da língua portuguesa e os *datasets*. Com esse material será possível continuar este trabalho avaliando outros modelos de aprendizado de máquina e realizando análises comparativas. Como etapas futuras pode-se, também, aumentar a lista de correção de termos da língua portuguesa.

AGRADECIMENTOS

Agradeço aos 25 voluntários que se disponibilizaram a rotular os tuítes e a Accenture por ter custeado a passagem área e o hotel para que eu pudesse viajar para o Rio de Janeiro e apresentar meu artigo no Webmedia.

[13]Repositório no Github com o código desenvolvido. https://github.com/yullidias/AutomaticIronyDetection/tree/tcc

REFERÊNCIAS

[1] Portal Action. [n.d.]. 5.1.2 - Cálculo e interpretação do p-valor. http://www.portalaction.com.br/inferencia/512-calculo-e-interpretacao-do-p-valor

[2] Yulli Alves, Ana Sanches, Daniel Dalip, and Ismael Silva. 2019. Automatic identification of irony: a case study on Twitter. *WebMedia '19: Proceedings of the 25th Brazillian Symposium on Multimedia and the Web*, 253–256. https://doi.org/10.1145/3323503.3360627

[3] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. 2011. *Modern Information Retrieval: The Concepts and Technology behind Search* (2nd ed.). Addison-Wesley Publishing Company, USA.

[4] Francesco Barbieri and Horacio Saggion. 2014. Modelling Irony in Twitter. In *Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Gothenburg, Sweden, 56–64. https://doi.org/10.3115/v1/E14-3007

[5] Krícia Helena Barreto. 2015. *Os memes e as interações sociais na internet: uma interface entre práticas rituais e estudos de face*. Ph.D. Dissertation. Tese (Doutorado em Linguística)—Universidade Federal de Juiz de Fora, Juiz ....

[6] Yonatan Belinkov and Yonatan Bisk. 2017. Synthetic and Natural Noise Both Break Neural Machine Translation. *CoRR* abs/1711.02173 (2017). arXiv:1711.02173 http://arxiv.org/abs/1711.02173

[7] Mondher Bouazizi and Tomoaki Ohtsuki. 2015. Opinion mining in Twitter: How to make use of sarcasm to enhance sentiment analysis. *2015 IEEE/ACM ASONAM*, 1594–1597.

[8] Jeunghyun BYUN, So-Young PARK, Seung-Wook LEE, and Hae-Chang RIM. 2009. Three-Phase Text Error Correction Model for Korean SMS Messages. *IEICE Transactions on Information and Systems* E92.D, 5 (2009), 1213–1217. https://doi.org/10.1587/transinf.E92.D.1213

[9] Diptesh Chatterhee. 2011. Correction of Noisy Sentences using a Monolingual Corpus. arXiv:cs.DL/1105.4318

[10] François Chollet. 2018. *Deep Learning with Python* (1nd ed.). Manning Publications Co., New York.

[11] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning* 20, 3 (1995), 273–297.

[12] Irene Cramer, Stefan Schacht, and Andreas Merkel. 2008. Classifying Number Expressions in German Corpora. In *Data Analysis, Machine Learning and Applications*. Springer, 553–560.

[13] Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Semi-supervised Recognition of Sarcastic Sentences in Twitter and Amazon. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning (CoNLL '10)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 107–116. http://dl.acm.org/citation.cfm?id=1870568.1870582

[14] Samridhi Dutta. 2017. Data Cleaning, Categorization and Normalization. https://dimensionless.in/data-cleaning-categorization-normalization/

[15] Elena Filatova. 2012. Irony and sarcasm: Corpus generation and analysis using crowdsourcing. In *In Language Resources and Evaluation Conf. , LREC2012*.

[16] Johannes Fürnkranz. 1998. A Study Using n-gram Features for Text Categorization.

[17] Pollyanna Gonçalves, Daniel Dalip, Julio Reis, Johnnatan Messias, Filipe Ribeiro, Philipe Melo, Leandro Rá Ujo, Fabrício Benevenuto, and Marcos Gonçalves. 2015. Bazinga! Caracterizando e Detectando Sarcasmo e Ironia no Twitter. In *BraSNAM - 5º Brazilian Workshop on Social Network Analysis and Mining*.

[18] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, et al. 2003. A practical guide to support vector classification. (2003).

[19] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. *An introduction to statistical learning*. Vol. 112. Springer.

[20] Daniel Jurafsky and James H. Martin. 2009. *Speech and Language Processing (2Nd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

[21] Arthur Kordon. 2010. *Applying Computational Intelligence* (1nd ed.). Springer-Verlag Berlin Heidelberg, New York. https://doi.org/10.1007/978-3-540-69913-2

[22] Jennifer Ling and Roman Klinger. 2016. An Empirical, Quantitative Analysis of the Differences Between Sarcasm and Irony. In *The Semantic Web*, Harald Sack, Giuseppe Rizzo, Nadine Steinmetz, Dunja Mladenić, Sören Auer, and Christoph Lange (Eds.). Cham, 203–216. https://doi.org/10.1007/978-3-319-47602-5_39

[23] Diana Maynard and M.A. Greenwood. 2014. Who cares about sarcastic tweets? Investigating the impact of sarcasm on sentiment analysis. *Proceedings of LREC* (01 2014), 4238–4243.

[24] Saif Mohammad, Parinaz Sobhani, and Svetlana Kiritchenko. 2016. Stance and Sentiment in Tweets. *ACM Transactions on Internet Technology* 17 (05 2016). https://doi.org/10.1145/3003433

[25] Douglas C. Montgomery and George C. Runger. 2009. *Estatística aplicada e probabilidade para engenheiros* (4nd ed.). LTC - Livros Técnicos e científicos Editora S.A., Rio de Janeiro.

[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[27] Ashwin Rajadesingan, Reza Zafarani, and Huan Liu. 2015. Sarcasm Detection on Twitter: A Behavioral Modeling Approach. *Proceedings of the 8th ACM WSDM*, 97–106. https://doi.org/10.1145/2684822.2685316
[28] Antonio Reyes, Paolo Rosso, and Tony Veale. 2013. A multidimensional approach for detecting irony in Twitter. *Language Resources and Evaluation* 47 (03 2013). https://doi.org/10.1007/s10579-012-9196-x
[29] S. S. Shapiro and M. B. Wilk. 1965. An Analysis of Variance Test for Normality (Complete Samples). *Biometrika* 52, 3/4 (1965), 591–611. http://www.jstor.org/stable/2333709
[30] Karthik Sundararajan and Anandhakumar Palanisamy. 2020. Multi-Rule Based Ensemble Feature Selection Model for Sarcasm Type Detection in Twitter. *Computational intelligence and neuroscience* 2020 (2020), 2860479. http://search-ebscohost-com.ez107.periodicos.capes.gov.br/login.aspx?direct=true&db=mdc&AN=32405293&lang=pt-br&site=ehost-live
[31] Rafael Rodrigues Troiani. 2010. File:Teste T Gráfico bicaudal.gif. https://commons.wikimedia.org/wiki/File:Teste_T_Gr%C3%A1fico_bicaudal.gif
[32] Twitter. [n.d.]. Como curtir um Tweet. https://help.twitter.com/pt/using-twitter/liking-tweets-and-moments
[33] Twitter. [n.d.]. Perguntas frequentes sobre Retweets. https://help.twitter.com/pt/using-twitter/retweet-faqs
[34] Twitter. [n.d.]. Sobre diferentes tipos de Tweets. https://help.twitter.com/pt/using-twitter/types-of-tweets
[35] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, CJ Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake Vand erPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1. 0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* 17 (2020), 261–272. https://doi.org/10.1038/s41592-019-0686-2
[36] Wikipédia. [n.d.]. Twitter. https://pt.wikipedia.org/wiki/Twitter
[37] Pratiksha Y. Pawar and Shravan Gawande. 2012. A Comparative Study on Different Types of Approaches to Text Categorization. *International Journal of Machine Learning and Computing* (01 2012), 423–426. https://doi.org/10.7763/IJMLC.2012.V2.158

Centro Federal de Educação Tecnológica de Minas Gerais

Curso de Engenharia de Computação

Avaliação do Trabalho de Conclusão de Curso

Aluna: Yulli Dias Tavares Alves
Título do trabalho: Automatic identification of Irony: a Case Study on Twitter
Data da defesa: 16/11/2020
Horário: 16:00
Local da defesa: <https://meet.google.com/xrb-rnww-ucc>

O presente Trabalho de Conclusão de Curso foi avaliado pela seguinte banca:

Daniel Hasan Dalip - Orientador

Departamento de Computação

Centro Federal de Educação Tecnológica de Minas Gerais

Lívia Maria Dutra

Departamento de Computação

Centro Federal de Educação Tecnológica de Minas Gerais

Ismael Santana Silva

Departamento de Computação

Centro Federal de Educação Tecnológica de Minas Gerais