

UNIVERSIDADE DE SÃO PAULO  
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA  
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

**Deteccção de sarcasmo em redes sociais  
utilizando DeBERTa**

Lucas Paiolla Forastiere

MONOGRAFIA FINAL  
MAC 499 — TRABALHO DE  
FORMATURA SUPERVISIONADO

Supervisor: Prof. Dr. Ricardo Marcondes Marcacini

São Paulo  
2017

*O conteúdo deste trabalho é publicado sob a licença CC BY 4.0  
(Creative Commons Attribution 4.0 International License)*

## Resumo

Lucas Paiolla Forastiere. **Deteção de sarcasmo em redes sociais utilizando DeBERTa**. Monografia (Bacharelado). Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2017.

[illegible]

**Palavras-chave:** Palavra-chave1. Palavra-chave2. Palavra-chave3.



# Abstract

Lucas Paiolla Forastiere. **Sarcasm detection in social media using decoding-enhanced BERT with disentangled attention**. Capstone Project Report (Bachelor). Institute of Mathematics and Statistics, University of São Paulo, São Paulo, 2017.

[illegible]

**Keywords:** Keyword1. Keyword2. Keyword3.



## Lista de Abreviaturas

CFT	Transformada contínua de Fourier ( <i>Continuous Fourier Transform</i> )
DFT	Transformada discreta de Fourier ( <i>Discrete Fourier Transform</i> )
EIIP	Potencial de interação elétron-íon ( <i>Electron-Ion Interaction Potentials</i> )
STFT	Transformada de Fourier de tempo reduzido ( <i>Short-Time Fourier Transform</i> )
ABNT	Associação Brasileira de Normas Técnicas
URL	Localizador Uniforme de Recursos ( <i>Uniform Resource Locator</i> )
IME	Instituto de Matemática e Estatística
USP	Universidade de São Paulo

## Lista de Símbolos

$\omega$	Frequência angular
$\psi$	Função de análise <i>wavelet</i>
$\Psi$	Transformada de Fourier de $\psi$

## **Lista de Figuras**

## **Lista de Tabelas**

## **Lista de Programas**



# Sumário

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Fundamentos e Trabalhos Relacionados</b>	<b>3</b>
2.1	Detecção de Sarcasmo . . . . .	3
2.2	Classificação binárias . . . . .	4
2.2.1	Métricas de avaliação . . . . .	4
2.3	Trabalhos Relacionados . . . . .	7
2.3.1	Abordagens Baseadas em Regras . . . . .	7
2.3.2	Conjuntos de Características . . . . .	9
2.3.3	Abordagens Baseadas em Métodos de Aprendizado . . . . .	11
2.3.4	Abordagens Baseadas em Contexto . . . . .	11
	<b>Referências</b>	<b>13</b>



# Capítulo 1

## Introduction

Sarcasm detection is an important aspect of many natural language processing (NLP) systems, with many implications in natural language understanding, dialog systems, and data mining. However, sarcasm detection is difficult because it is infrequent in many conversations and, many times, it is difficult even for humans to discern.

Many studies have been made in the area and many datasets have been proposed with either *balanced* or *unbalanced* data. Also many of these datasets use humans to annotate sarcastic statements.

In this paper, we use the Self-Annotated Reddit Corpus (SARC), which is a large corpus for sarcasm detection created using Reddit posts to get labels automatically, to train, evaluate, and compare many NLP models.



## Capítulo 2

# Fundamentos e Trabalhos Relacionados

### 2.1 Detecção de Sarcasmo

De acordo com o dicionário Dicio, sarcasmo é uma zombaria que busca ofender, enquanto ironia é a ação de dizer o oposto do que se deseja expressar. Ainda segundo ele, a diferença entre esses dois termos se dá no fato de que sarcasmo é um dito ácido que pode ou não ser expresso por meio de uma ironia e essa, por sua vez, pode ou não ser utilizada para ofender. [DICIO, 2022b](#); [DICIO, 2022a](#)

[JOSHI, BHATTACHARYYA \*et al.\*, 2016](#)

(R. Giora, On Irony and Negation)

(H. Paul Grice. 1975. Logic and Conversation)

(Irony and Sarcasm: Corpus Generation and Analysis Using Crowdsourcin)

O termo detecção de sarcasmo refere-se à determinação de se há ou não sarcasmo em uma porção de texto verbal. E o termo detecção automática de sarcasmo refere-se a métodos computacionais de se resolver o problema acima mencionado. Computacionalmente, podemos definir esse problema como uma **classificação binária**, termo explicado mais a frente.

Entretanto, na literatura é bastante comum também se definir detecção de sarcasmo como a determinação de se há ou não sarcasmo ou ironia verbal em uma porção de texto verbal. Portanto, em geral, ao se falar de sarcasmo, a literatura engloba tanto sarcasmo quanto ironia como se fossem a mesma coisa. Este texto também não fará discriminação entre sarcasmo ou ironia.

Em geral, esse problema é difícil, pois, por vezes, nem humanos conseguem perceber essa figura de linguagem (e.g. “*Oba, hoje está tão ensolarado, que vontade de ir para a escola.*”). Além disso, o contexto tende a importar muito. Muitas características externas ao texto podem servir para discriminar se uma pessoa está ou não sendo sarcástica. Entre

alguns exemplos estão intonação, locutor, interlocutor, conhecimento prévio sobre a fala, tempo e espaço em que se fala e elementos não verbais. WALLACE *et al.*, 2014

(Humans Require Context to Infer Ironic Intent (so Computers Probably do, too))

## 2.2 Classificação binárias

O problema de detecção de sarcasmo pode ser tratado como uma classificação binária. Esse tipo de problema é muito estudado no campo do aprendizado de máquina e envolve classificar os elementos de um conjunto em dois grupos chamados de classes. Por classificar entende-se a determinação de um elemento do conjunto entre pertencente à primeira ou segunda classe.

No caso da detecção de sarcasmo, os elementos são os pedaços de texto e as duas classes são *ser sarcástico* e *não ser sarcástico*. Em termos matemáticos, tem-se um conjunto de exemplos denotado pela matriz  $X$  de dimensões  $n \times m$ , onde  $n$  é o número de exemplos e  $m$  é o número de características que se usa para descrever cada um desses exemplos. Cada linha de  $X$  é denotada por  $x^{(i)}$  e chamada de *exemplo  $i$*  ou *instância  $i$* .

A linha  $x^{(i)}$  é um vetor de  $m$  posições em que cada posição é uma característica que descreve a entrada. Por exemplo, na detecção de sarcasmo, o primeiro valor pode ser a quantidade de palavras lidas, o segundo valor pode ser a quantidade de caracteres, a terceira pode ser a soma de positividade do texto (definida por algum critério específico) e assim por diante. É importante que a mesma posição em diferentes instâncias tenha o mesmo significado. Portanto, caso se defina que a primeira posição represente o número de palavras, todas as instâncias devem seguir essa regra.

Cada instância  $i$  pertence ou à primeira classe ou à segunda, modela-se isso por um valor  $y^{(i)}$  chamado de  $i$ -ésimo rótulo (do inglês, *label*). E escreve-se  $y^{(i)} \in \{0, 1\}$ , onde cada valor representa uma das classes. No caso de sarcasmo, pode-se representar por *não sarcasmo* o valor 0 e *sarcasmo*, o valor 1. Com esses valores  $y^{(i)}$  constrói-se um vetor  $y$  onde  $y_i = y^{(i)}$ .

Ao final, o problema é descobrir uma função  $f$  que mapeie bem  $X$  em  $y$ . Note, entretanto, que  $n$ , o número de exemplos, é possivelmente infinito, pois sempre se pode achar novos exemplos e testar se  $f$  os mapeia corretamente. Ou seja, tenta-se achar essa função  $f$  conhecendo parcialmente o conjunto das instâncias.

### 2.2.1 Métricas de avaliação

Para saber se a função  $f$  mapeia bem  $X$  em  $y$ , é preciso definir uma métrica de avaliação, que é uma função matemática bem definida que indica quanto de erro se comete em determinado conjunto  $X$ .

Dado um conjunto de entrada  $X$  com  $n$  instâncias, seja  $y$  o vetor que representa se cada instância é ou não sarcástica. Seja então  $f$  a função sob avaliação (ou seja, a função de modelagem que recebe os textos e determina se são ou não sarcásticos). Seja  $\hat{y} = f(X)$  os resultados gerados por essa função.

Note que para cada exemplo  $i$ , podemos ter duas opções:

1.  $\hat{y}_i = y_i$  (*acerto*)
2.  $\hat{y}_i \neq y_i$  (*erro*)

Além disso, pode-se note que o par  $(y_i, \hat{y}_i)$  pode ter quatro valores:

1. (1, 1), caso *verdadeiro positivo* (também conhecido como *true positive* e denotado por  $tp$ ).
2. (0, 1), caso *falso positivo* (também conhecido como *false positive* e denotado por  $fp$ ).
3. (0, 0), caso *verdadeiro negativo* (também conhecido como *true negative* e denotado por  $tn$ ).
4. (1, 0), caso *falso negativo* (também conhecido como *false negative* e denotado por  $fn$ ).

Definidos esses valores, pode-se definir algumas métricas que ajudam a perceber quão boa a função  $f$  é. Dessa forma, pode-se comparar duas funções  $f$ .

**Acurácia** Definida simplesmente como a quantidade total de acertos dividido pela quantidade total de instâncias. Seja  $n'$  a quantidade de acertos, então:

$$\text{Acurácia} = \frac{n'}{n}$$

É comum também definir a acurácia em termos dos vamos acima listados:

$$\text{Acurácia} = \frac{tp + tn}{tp + tn + fp + fn}$$

Note que  $tp + tn$  representa justamente quando  $\hat{y}_i = y_i$  e  $fp + fn$ ,  $\hat{y}_i \neq y_i$ .

**Precisão** Definida como:

$$\text{Precisão} = \frac{tp}{tp + fp}$$

é a taxa de verdadeiros positivos em relação a todos os positivos. Portanto, mede a fração de quantos positivos foram acertados em relação a todos os que existiam no conjunto observado.

**Revocação** Muitas vezes chamada por seu nome em inglês, *recall*, é definida como:

$$\text{Revocação} = \frac{tp}{tp + fn}$$

é a taxa de verdadeiros positivos em relação todas as instâncias previstas como positivas. Portanto, a fração entre todos as instâncias previstas como positivas e as instâncias que realmente eram positivas.

A precisão e a revocação guardam uma relação chave entre si e geralmente se quer ter um bom balanço entre as duas. Observe essa relação a partir do seguinte exemplo.

Comece com a função

$$f(x^{(i)}) = 1 \quad \forall i \quad (2.1)$$

Como ela sempre considera a entrada como positiva, então os únicos casos existentes serão de verdadeiros positivos e falso positivos. Logo:

$$\text{Precisão} = \frac{tp}{tp + fp} = \frac{tp}{tp + (n - tp)} = \frac{tp}{n}$$

e

$$\text{Revocação} = \frac{tp}{tp + fn} = \frac{tp}{tp + 0} = 1$$

Portanto, a revocação terá o maior valor possível, enquanto a precisão será exatamente a fração de valores positivos que o conjunto observado possui. Como, semanticamente, a revocação é penalizada sempre que um elemento é positivo, mas foi previsto como negativo, então faz sentido que se tenha uma revocação de 100%, já que não se comete esse tipo de erro.

Agora, tome a função

$$f(x^{(i)}) = 0 \quad \forall i \quad (2.2)$$

Como ela sempre considera a entrada como negativa, então os únicos casos existentes serão de verdadeiro negativo e falso negativo. Logo:

$$\text{Precisão} = \frac{tp}{tp + fp} = \frac{0}{0 + fp} = 0$$

e

$$\text{Revocação} = \frac{tp}{tp + fn} = \frac{0}{0 + fn} = 0$$

Portanto, como a função nunca tenta marcar uma instância como positiva, ela nunca acerta os verdadeiros positivos e comete 100% de erro.

**Métrica  $F_1$**  Por esses e outros motivos, é interessante agregar a precisão e revocação em um único valor, que permite fácil comparação entre dois modelos. É possível utilizar uma média simples desses dois valores, mas é muito mais comum a utilização da métrica  $F_1$  (ou, do inglês,  $F_1$  score).

Ela é definida como:

$$F_1 = 2 \cdot \frac{\text{precisão} \cdot \text{revocação}}{\text{precisão} + \text{revocação}}$$



## 2.3 Trabalhos Relacionados

Nessa seção abordar-se-á os trabalhos já realizados na área e as diferentes abordagens. Pode-se caracterizar os trabalhos por três principais tipos de abordagens: as baseadas em regras, baseadas em métodos de aprendizado e baseadas em contexto. Além disso, falar-se-á um pouco sobre as principais características extraídas dos textos que têm sido utilizadas para melhorar a eficácia das soluções.

### 2.3.1 Abordagens Baseadas em Regras

Abordagens baseadas em regras são aquelas que usam regras fixas para determinar se uma sequência de palavras contém ou não ironia. Para criar essas regras, várias características do texto podem ser utilizadas, como as classes sintáticas das palavras, se são palavras de cunho positivo ou negativo, se há a presença ou não de certas palavras em uma ordem, entre qualquer outro tipo de regra imaginável.

Essa abordagem é computacional, porque as regras formam um algoritmo que pode ser implementado por uma linguagem de computação. Dessa forma, qualquer sequência de palavras pode ser dada como entrada para o algoritmo e ele retornará se ele acredita que essa sequência contém ou não sarcasmo.

A principal vantagem desse tipo de abordagem é que ela, além de detectar o sarcasmo, ajuda a estudá-lo. Ao criar uma regra do tipo *se o texto possui essas características, então ele é sarcástico* se cria uma explicação para as principais características presentes em um texto sarcástico.

Entretanto, as metodologias utilizadas são bastante específicas para cada tipo de texto e cada conjunto de dados. Por exemplo, as regras criadas para um determinado conjunto da rede social Twitter podem não funcionar em outras redes sociais como o Facebook, o Instagram ou o WhatsApp, pois cada uma dessas redes sociais possui um contexto e formato de conversação muito diferente. Portanto, são modelos que nos permitem explicar o processo de decisão, mas não permitem fácil generalização.

VEALE e HAO, 2010 investigam em seu artigo sequências da forma “as \* as a \*\*” (“tão \* quanto um(a) \*\*”), consideradas como analogias entre o que os autores chamam de *base* (*ground*) e *meio* (*vehicle*). Eles utilizam a API do Google para coletar 45021 instâncias do padrão “*about as \* as \*\**” e filtram os resultados na mão para chegar em 20299 instâncias que de fato são consideradas analogias. Eles então anotam manualmente os rótulos para essas instâncias e encontraram que 15502 casos (76%) são irônicos e apenas 4797 (24%) são não irônicos.

Então, dado uma sequência “as \* as a \*\*”, os autores utilizam mecanismos de busca na rede como a API do Google para distinguir entre três casos: os casos em que essa sequência nunca foi usada como “*about*”; aqueles que já foram, mas não frequentemente; e aqueles que são frequentemente usados com essa marcação. Segundo os próprios autores, essas três categorias proveem, respectivamente, evidência fraca contra ironia, evidência fraca a favor da ironia e evidência forte para ironia. Ou seja, o caso em que temos mais certeza de que é uma ironia é o caso em que a sequência é frequentemente utilizada junto com a palavra “*about*”.

Assim sendo, **VEALE e HAO, 2010** criam um sequência de nove passos para classificar uma frase desse tipo entre as classes *ironia* e *não ironia*. Esses passos são bastante claros e precisos, podendo ser, portanto, implementados por um programa de computador e caracterizando, por conseguinte, uma detecção automática de sarcasmo.

Nessas regras, os autores utilizam o fato de que analogias mais frequentemente utilizadas são menos prováveis de serem irônicas, pois a ironia é utilizada bastante a criatividade do locutor para fazer analogias não usuais.

Devido à natureza das regras propostas pelos autores, eles conseguem medir a precisão e revocação obtida por cada uma das regras. No geral, o modelo atinge uma acurácia de 88%, o que é um valor bastante alto para esse tipo de problema. Entretanto, note-se que os autores se limitaram a um escopo muito fechado (das frases do tipo “*as \* as a \**”).

**MAYNARD e GREENWOOD, 2014** exploram o uso de regras baseadas em hashtags presentes em postagens da rede social Twitter (<https://twitter.com/>) e sua aplicação para detecção de sarcasmo em um contexto mais geral de análise de sentimento.

Em seu artigo, eles utilizam um algoritmo de tokenização de hashtags para transformá-las em palavras com as quais eles conseguem extrair informação. No caso, eles utilizam as palavras contidas nas hashtags para avaliar se o sentimento é positivo ou negativo e inverter o sentido original da frase caso detectem sarcasmo nas hashtags. Por exemplo, a hashtag *#notreally* é tokenizada em *not* e *really* e identificada como uma tag sarcástica de acordo com regras definidas pelos autores.

Então, eles aplicam cinco regras que podem determinar o sentimento de uma postagem como *negativo* ou *positivo*, ou então trocar o sentimento pré-determinado da postagem. Os autores, portanto, fazem uso da detecção de sarcasmo utilizando hashtags para resolver um outro problema mais geral que é definir o sentimento predominante de um texto.

Como ponto negativo de sua técnica, está o fato de que nem sempre o tokenizador de hashtags funciona de forma adequada. Por exemplo, *#greatstart* pode ser dividida de duas formas: *greats* e *tart* ou *great* e *start*. Um ser humano provavelmente saberia que a segunda opção é a mais provável, mas o algoritmo não sabe fazer essa distinção e acaba ficando com o primeiro conjunto e palavras. Apesar disso, esse sistema de tokenização possui um *F1* de 97,25% e os autores obtiveram 91% de precisão e revocação no conjunto de dados utilizado por eles.

**BHARTI et al., 2015** apresentam duas abordagens baseadas em regras. A primeira é através da análise morfossintática (do inglês, *Part-of-Speech Tagging*) e criação de árvores sintáticas (do inglês, *parse-trees*) que identificam a positividade do sentimento e situação predominantes em tweets. A segunda é através da análise sintática de tweets que começam com interjeições.

Em sua primeira abordagem, os autores utilizam dicionários e algoritmos pré-criados para encontrar as classes morfológicas e sintáticas das palavras. Então, baseando-se nisso, eles utilizam um algoritmo criado por eles para encontrar um valor de positividade para o sentimento do texto e para a situação retratada e, com esses valores, caso seus valores tenham sinais contrários (sentimento positivo e situação negativa ou vice-versa), eles consideram o texto como sarcástico. Por exemplo, em “*I hate Australia in cricket, because*

*they always win*” (“eu odeio a Austrália no críquete, porque eles sempre ganham”), as palavras “*I hate*” apresentam um sentimento negativo, enquanto as palavras “*they always win*” retratam uma situação positiva, e, portanto, essa frase seria classificada como sarcástica em sua primeira abordagem.

Em sua segunda abordagem, os autores utilizam novamente algoritmos de *POS Tagging* pré-criados para achar as classes morfológicas e sintáticas de palavras em tweets que começam com interjeições, como *wow*, *oh*, *wow*, *aha*, *yay*, *yeah*, *nah*, etc. Em seu algoritmo, eles classificam como sarcásticos os textos que começam por interjeições e possuem um adjetivo ou advérbio imediatamente após a interjeição, ou então possuem, em algum após a interjeição, ou um advérbio seguido de adjetivo ou um adjetivo seguido de um substantivo ou um advérbio seguido de um verbo. Essa abordagem é bastante interessante, pois ela é bastante simples e, ainda assim, consegue um ótimo resultado de 0.90  $F_1$  no subconjunto de tweets marcado com a hashtag *sarcasm*.

### 2.3.2 Conjuntos de Características

Antes de prosseguir com as abordagens, essa seção mostra um pouco dos principais conjuntos de características usadas por essas abordagens. Abordagens baseadas em regras utilizam determinadas características como principalmente as classes morfosintáticas das palavras. Entretanto, as abordagens listadas a seguir fazem uso de conjuntos muito maiores de características retiradas do texto e vários trabalhos foram realizados para expandir a quantidade de características possíveis de se utilizar.

Como dito anteriormente, ao criar a matriz de exemplos de entrada  $X$ , cada coluna representa uma característica do texto. Dessa forma, o texto fica agregado em um conjunto de valores numéricos que são acessíveis para a máquina utilizar e comparar. Ao invés de se comparar dois textos distintos com números de caracteres e palavras diferentes, utiliza-se um algoritmo para extrair características (do inglês, “*features*”) comuns entre os textos e compará-los por meio dessas características.

Um exemplo é o método chamado *bag-of-words* (em tradução literal “sacola-de-palavras”) que cria um vetor com, por exemplo, 512 posições de valores naturais onde cada posição do vetor representa quantas vezes uma determinada palavra apareceu no texto. Cada posição do vetor é associada a uma palavra e, muitas vezes, se utiliza uma outra posição especial para denotar qualquer palavra que não está no dicionário de palavras selecionadas previamente. Por exemplo, a primeira posição pode representar quantas vezes a palavra “*a*” aparece no texto, a segunda posição pode representar quantas vezes a palavra “*the*” aparece, e assim por diante. Se a palavra “Lucas” for lida, então será adicionado um à última posição do vetor, que é utilizado para contar qualquer palavra que não é uma das outras 511.

(imagem ilustrando BOW)

Esse tipo de abordagem fica entre o baseado em regras e abordagens baseadas em aprendizado no quesito de explicabilidade dos modelos. Isso porque os autores, ao invés de criar regras específicas, utilizam métodos de extração de características, criadas especificamente com o intuito de fazer a detecção de sarcasmo.

A vantagem dessa abordagem é que geralmente os modelos são mais generalistas do que os baseados em regras, que são bastante fixas e não se adaptam bem a diferentes conjuntos de dados. Nesse caso, as mesmas metodologias de extração de características podem ser aplicadas em um conjunto de dados de textos extraídos de contextos diferentes (como redes sociais, análises de produtos vendidos online, aplicativos de troca de mensagens ou falas transcritas).

Entretanto, como desvantagem, o fator manual ainda é muito presente nesse tipo de abordagem, assim como os métodos baseados em regras. Ao criar um modelo, um determinado conjunto de características é escolhido *a priori* arbitrariamente, assim como era o caso das regras no primeiro tipo de abordagem.

REYES, ROSSO e BUSCALDI, 2012 exploram quatro grupos de características dentro de um contexto mais geral de detecção de humor e ironia. Esses grupos são: ambiguidade, através dos níveis estrutural, morfosintático e semântico; polaridade, através de palavras que possuam semântica positiva ou negativa; imprevisibilidade, através de desbalanceamentos contextuais entre o significado literal das palavras; e os cenários emocionais, através das emoções passadas por cada palavras.

LIEBRECHT *et al.*, 2013 utilizam uni-, bi- e trigramas como características do texto. Um n-grama é uma sequência de n palavras do texto. Portanto, nesse tipo de característica, o unigrama seria o *bag-of-words* clássico citado acima, com as contagens de cada palavra, já o bigrama se trata das contagens de duas palavras seguidas. Por exemplo, sempre que as palavras “so nice” aparecerem em sequência, adiciona-se um à coluna que se refere a essa sequência.

REYES, ROSSO e VEALE, 2013 utilizam uma abordagem parecida com REYES, ROSSO e BUSCALDI, 2012, com quatro grupos de características: assinaturas, imprevisibilidade, estilo e cenários emocionais. Cada um dos grupos foca em várias características. Dentro das assinaturas, por exemplo, há o foco em pontuações específicas, emojis, citações e palavras em maiúsculas; há o foco em marcas implícitas que geram oposição entre partes do texto; e há o foco na compressão temporal, que identifica oposição temporal entre elementos relacionados.

A grande diferença desse trabalho em relação aos demais está, entretanto, no grupo de características de estilo. Esse grupo tenta modelar o estilo de escrita do texto para, assim, fazer a diferenciação entre estilos sarcásticos e não sarcásticos. Para fazer isso, eles utilizam três tipos de sequências textuais: n-gramas no nível de caracteres (ao invés de palavras) (abreviado pelos autores por *c-grams*), *skip-grams* (*s-grams*) e *skip-grams* de polaridade (*ps-grams*).

Os *c-grams* capturam a frequência de sequências de informação morfológica como sufixos e prefixos (como *-ly*, *-ing*, *dis-*, *un-*). Os *s-grams* funcionam como n-gramas, mas com alguns pulos entre palavras e são utilizados para obter relações entre palavras não adjacentes no texto. Por exemplo, na sentença “Hoje está chovendo, quero tanto ir à praia”, um bigrama é representado pela sequência “Hoje está”, enquanto um bigrama com pulo de uma palavra seria “Hoje chovendo”. Por fim, os *ps-grams* criam uma sequência de rótulos de positividade baseado nos *s-grams*. Por exemplo, “Hoje chovendo” seria rotulado como “pos-neg” (positivo e negativo).

BARBIERI *et al.*, 2014, por sua vez, usam sete grupos de características léxicas que abstraem o uso de termos específicos e tornam o processo de detecção de sarcasmo mais genérico e fácil de implementar, não modelando padrões de palavras como em trabalhos anteriores. Esses grupos são: frequência, que mede o quão comum é cada palavra utilizada no texto; escrito-falado (da tradução literal de *written-spoken* utilizado pelos autores em seu artigo), que funciona como a frequência, mas a frequência que a palavra é usada em diálogos escritos ou falados; intensidade, que mede o uso de adjetivos e advérbios para aumentar ou diminuir a intensidade semântica de outras palavras no texto; estrutura, que mede, por exemplo, o número de caracteres usados, o número de palavras, o tamanho médio das palavras, o número de verbos, de substantivos, o número total de pontuações, o número de risadas, o número de vírgulas, pontos finais, sinais de exclamação, o número de emojis (ou emoticons), o número de organizações, pessoas, títulos e datas citados pelos texto; sentimento, que mede valores de positividade entre as palavras e usa algumas métricas como soma dos valores positivos, soma dos negativos, média da diferença entre positivos e negativos, entre outros; sinônimos, que calcula métricas relacionadas aos sinônimos das palavras usadas no texto, como a quantidade de sinônimos que uma determinada palavra tem e que possuem frequência de utilização menor do que ela; e ambiguidade, que mede a quantidade de sentidos possíveis que uma palavra pode ter.

JOSHI, SHARMA *et al.*, 2015 propõem uma nova abordagem baseada em uma teoria linguística chamada de incongruência de contexto. Eles se baseiam no fato de que o tempo para um ser humano processar um texto sarcástico depende do grau de incongruência presente nele e criam características o que chamam de incongruências explícitas e implícitas. A primeira é caracterizada por palavras de sentimentos opostos e a segunda é caracterizada por frases de sentimento implícito, que não se pode detectar através de uma única palavra.

MISHRA *et al.*, 2016 seguem por uma abordagem bastante diferente das demais, que utiliza características baseadas no movimento produzido pelos olhos de pessoas lendo textos sarcásticos. Algumas das características propostas por eles são a fixação da visão em pontos do texto e saltos mais rápidos entre duas porções do texto do que em relação às demais. Além disso, eles utilizam um grafo de informação estrutural ao utilizar as palavras como vértices do grafo e os saltos de visão entre as palavras como arestas.

### 2.3.3 Abordagens Baseadas em Métodos de Aprendizado

Como já mencionado, a modelagem clássica de um problema de detecção de sarcasmo é utilizando uma matriz de características  $X$  e um vetor de rótulos  $y$ .

### 2.3.4 Abordagens Baseadas em Contexto



## Referências

- [BARBIERI *et al.* 2014] Francesco BARBIERI, Horacio SAGGION e Francesco RONZANO. “Modelling sarcasm in Twitter, a novel approach”. Em: *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Baltimore, Maryland: Association for Computational Linguistics, jun. de 2014, pgs. 50–58. DOI: [10.3115/v1/W14-2609](https://doi.org/10.3115/v1/W14-2609). URL: <https://aclanthology.org/W14-2609> (citado na pg. 11).
- [BHARTI *et al.* 2015] Santosh Kumar BHARTI, Korra Sathya BABU e Sanjay Kumar JENA. “Parsing-based sarcasm sentiment recognition in twitter data”. Em: *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. 2015, pgs. 1373–1380. DOI: [10.1145/2808797.2808910](https://doi.org/10.1145/2808797.2808910) (citado na pg. 8).
- [DICIO 2022a] DICIO. *Ironia - Dicio, Dicionário Online de Português*. URL: <https://www.dicio.com.br/ironia/> (acesso em 19/09/2022) (citado na pg. 3).
- [DICIO 2022b] DICIO. *Sarcasmo - Dicio, Dicionário Online de Português*. URL: <https://www.dicio.com.br/sarcasmo/> (acesso em 19/09/2022) (citado na pg. 3).
- [JOSHI, BHATTACHARYYA *et al.* 2016] Aditya JOSHI, Pushpak BHATTACHARYYA e Mark James CARMAN. *Automatic Sarcasm Detection: A Survey*. 2016. DOI: [10.48550/ARXIV.1602.03426](https://doi.org/10.48550/ARXIV.1602.03426). URL: <https://arxiv.org/abs/1602.03426> (citado na pg. 3).
- [JOSHI, SHARMA *et al.* 2015] Aditya JOSHI, Vinita SHARMA e Pushpak BHATTACHARYYA. “Harnessing context incongruity for sarcasm detection”. Em: jul. de 2015. DOI: [10.3115/v1/P15-2124](https://doi.org/10.3115/v1/P15-2124) (citado na pg. 11).
- [LIEBRECHT *et al.* 2013] Christine LIEBRECHT, Florian KUNNEMAN e Antal van den BOSCH. “The perfect solution for detecting sarcasm in tweets #not”. Em: *WASSA@NAACL-HLT*. 2013 (citado na pg. 10).
- [MAYNARD e GREENWOOD 2014] Diana MAYNARD e Mark GREENWOOD. “Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis.” Em: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14)*. Reykjavik, Iceland: European Language Resources Association (ELRA), mai. de 2014, pgs. 4238–4243. URL: [http://www.lrec-conf.org/proceedings/lrec2014/pdf/67\\_Paper.pdf](http://www.lrec-conf.org/proceedings/lrec2014/pdf/67_Paper.pdf) (citado na pg. 8).

- [MISHRA *et al.* 2016] Abhijit MISHRA, Diptesh KANOJIA, Seema NAGAR, Kuntal DEY e Pushpak BHATTACHARYYA. “Harnessing cognitive features for sarcasm detection”. Em: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, ago. de 2016, pgs. 1095–1104. DOI: [10.18653/v1/P16-1104](https://doi.org/10.18653/v1/P16-1104). URL: <https://aclanthology.org/P16-1104> (citado na pg. 11).
- [REYES, ROSSO e BUSCALDI 2012] Antonio REYES, Paolo Rosso e Davide BUSCALDI. “From humor recognition to irony detection: the figurative language of social media”. Em: *Data & Knowledge Engineering* 74 (2012). Applications of Natural Language to Information Systems, pgs. 1–12. ISSN: 0169-023X. DOI: <https://doi.org/10.1016/j.datak.2012.02.005>. URL: <https://www.sciencedirect.com/science/article/pii/S0169023X12000237> (citado na pg. 10).
- [REYES, ROSSO e VEALE 2013] Antonio REYES, Paolo Rosso e Tony VEALE. “A multidimensional approach for detecting irony in twitter”. Em: *Language Resources and Evaluation* 47.1 (mar. de 2013), pgs. 239–268. ISSN: 1574-0218. DOI: [10.1007/s10579-012-9196-x](https://doi.org/10.1007/s10579-012-9196-x). URL: <https://doi.org/10.1007/s10579-012-9196-x> (citado na pg. 10).
- [VEALE e HAO 2010] Tony VEALE e Yanfen HAO. “Detecting ironic intent in creative comparisons”. Em: *Proceedings of the 2010 Conference on ECAI 2010: 19th European Conference on Artificial Intelligence*. NLD: IOS Press, 2010, pgs. 765–770. ISBN: 9781607506058 (citado nas pgs. 7, 8).
- [WALLACE *et al.* 2014] Byron C. WALLACE, Do Kook CHOE, Laura KERTZ e Eugene CHAR-  
NIAK. “Humans require context to infer ironic intent (so computers probably do, too)”. Em: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Baltimore, Maryland: Association for Computational Linguistics, jun. de 2014, pgs. 512–516. DOI: [10.3115/v1/P14-2084](https://doi.org/10.3115/v1/P14-2084). URL: <https://aclanthology.org/P14-2084> (citado na pg. 4).