Irony and Sarcasm detection with BERT (Bidirectional Encoder Representation from Transformer) model

Detecção de sarcasmo e ironia utilizando o modelo BERT (Bidirectional Encoder Representation from Transformer)

Abstract: Artificial intelligence is increasingly present in people's daily lives, machine learning algorithms have allowed the creation of applications involving image and text, interacting more and more with human beings. However, certain tasks are still challenging for machines due to the high degree of subjectivity. Sarcasm and irony are complex forms of expressions widely used by humans in their communication, such forms of expression are an example of a barrier for machines and even humans to be able to interpret comments, their understanding demands total contextualization about the environment in which the comment is made. Ways to better represent the text, allowing its interpretation through algorithms have evolved a lot and if before they could not represent the context present in a sentence, now through advances as in the pre-trained text representation model BERT made available by Google, it is possible to extract the context of a sentence, distinguishing. Thus, this work presents a method that uses the pre-trained model BERT, optimizes it to better represent comments present in the Reddit social network and trains an additional layer of neural network to detect irony and sarcasm in your comments. The method was implemented and achieved a result of 70% accuracy in the direct classification about the presence or not of sarcasm and irony in comments. The method was also evaluated against different benchmarks in the following task: given two comments on a post of which only one contains sarcasm and irony identifies it. In this second evaluation, the method obtained an accuracy of 78.7% and was above benchmarks that also used machine learning, but with simpler techniques for text representation (accuracy between 70% and 75%). When compared against a human benchmark, that is, one human and a group of five humans performing the same activity, the method performed worse (accuracy of 81.6% and 92% respectively).

Key*word*s: Machine Learning, BERT, Text Representation, Irony, Sarcasm.

Resumo: Inteligência artificial está cada vez mais presente no dia a dia das pessoas, algoritmos de aprendizado de máquina têm permitido a criação de aplicações que envolvem imagem e texto, permitindo cada vez mais interação com o ser humano. Porém determinadas tarefas ainda são desafiadoras para as máquinas devido ao alto grau de subjetividade. Sarcasmo e ironia são formas complexas de expressão muito utilizadas por humanos em sua comunicação, tais formas são exemplos de barreiras para que máquinas e mesmo humanos consigam interpretar comentários, sua compreensão demanda total contextualização sobre o ambiente no qual o comentário é realizado. Formas para melhor representar o texto têm evoluído, permitindo a sua interpretação através de algoritmos, se antes era possível representar o contexto presente em uma sentença de maneira muito pobre, atualmente com o modelo de representação de texto pré-treinado BERT disponibilizado pelo Google se tem a intenção de extrair o contexto de uma sentença de forma mais rica. Neste trabalho é apresentado um método que utiliza o modelo pré-treinado BERT, aperfeiçoa-o para melhor representar comentários presentes na rede social Reddit e treina uma camada adicional de rede neural para detectar ironia e sarcasmo em

seus comentários. O método foi implementado e alcançou o resultado de 70% de acurácia na classificação direta sobre a presença ou não de sarcasmo e ironia em comentários (ou seja, o método acerta sete de cada dez comentários classificados). O método também foi avaliado contra *benchmarks* distintos na seguinte tarefa: dados dois comentários de uma postagem, dos quais apenas um deles contém sarcasmo e ironia, identifique-o. Nesta segunda avaliação, o método obteve uma acurácia de 78,7% e ficou acima de *benchmarks* que também utilizaram aprendizado de máquina, porém com técnicas mais simples para representação de texto (acurácias entre 70% e 75%). Quando comparado contra *benchmark* humano, ou seja, um humano ou um grupo de cinco humanos realizando a mesma atividade, o método teve pior desempenho (acurácias humanos de 81,6% e 92%, respectivamente).

Palavras-chave: Aprendizado de Máquina, BERT, Representação de Texto, Ironia, Sarcasmo.

## 1. INTRODUÇÃO

A identificação automática de sarcasmo e ironia em texto tem se mostrado relevante devido ao impacto causado na tarefa de análise de sentimento, uma vez que a natureza figurativa de sarcasmo e ironia implica na inversão de polaridade do sentimento proposto pelo autor do texto (LIU, 2010). Assim, a detecção automática de sarcasmo e ironia acabou se tornando um problema de pesquisa dentro da comunidade de Processamento de Linguagem Natural (PLN), ou, em inglês, *Natural Language Processing* (NLP), e diversos métodos estão sendo propostos para classificar se um texto contém ou não tais elementos. Um dos primeiros trabalhos conhecidos para detecção de sarcasmo e ironia foi proposto por Tepperman, Traum e Narayanan (2006), no qual a detecção é realizada em mensagens transcritas de diálogos de um *call center*. Muitas abordagens distintas são encontradas na literatura, como sistemas baseados em regras, analise léxica, aprendizado de máquina supervisionado e não supervisionado. As bases de dados utilizadas para estudo também são as mais distintas, com maior frequência em bases extraídas do Twitter e fóruns de discussões. Toda esta sinergia de métodos distintos trouxe muita inovação para o desafio (JOSHI; BHATTACHARYYA; CARMAN, 2016).

A NLP, subárea do tópico inteligência artificial, é definida por Goyal, Pandey e Jain (2018) como a habilidade de um computador ou sistema entender profundamente a linguagem humana e processá-la da mesma forma que um humano, a mesma tem passado por grande evolução nos últimos anos. Um de seus grandes desafios é criar técnicas para extrair as características do texto e criar representações estruturadas que mantenham a informação contida nele. Com estas representações estruturadas é possível alimentar algoritmos de aprendizado de máquina e resolver tarefas de NLP (BROWNLEE, 2017).

Grande parte do sucesso em tarefas que envolvem NLP está relacionado com a qualidade em que o texto é representado de forma estruturada, quanto mais informações são obtidas e utilizadas na representação do texto, melhores são os resultados, em razão disso, muito esforço está sendo aplicado nesta área.

Uma técnica muito comum para a extração de características do texto é a *bag-of-words*. Esta técnica analisa todas as palavras presentes em um determinado texto e cria um vetor para representá-lo, no qual cada índice informa o número de vezes que cada uma de suas palavras se repete (GOLDBERG; HIRST, 2017). A *bag-of-words* viabilizou muitas

aplicações de NLP, teve algumas evoluções, porém por não gerar contexto entre as palavras dentro de um mesmo texto, muita informação é desprezada.

Para obtenção de melhores resultados em tarefas de NLP, a pesquisa por melhores representações de texto continua intensa. Iniciaram-se então pesquisas que testaram a aplicação de algoritmos de aprendizado de máquina profundo para alcançar uma melhor representatividade de texto.

Para otimizar a representação de texto, utilizando o maior volume de dados possível, foi desenvolvida a técnica conhecida como *Transfer Learning*, a qual foca na resolução do problema comum de insuficiência de dados para aplicação de aprendizado de máquina. *Transfer Learning* tem por objetivo levar o conhecimento de um domínio origem para um domínio alvo, desprezando a premissa de que dados de treino e dados de teste devem ser independentes e identicamente distribuídos. Esta técnica apresentou um ótimo efeito em muitas tarefas, as quais, anteriormente, eram difíceis obterem melhores resultados devido à insuficiência de dados para treinamento (TAN et al., 2018).

Com a evolução do *Transfer Learning*, modelos pré-treinados começaram a ser desenvolvidos para finalidades genéricas, nas quais existem dados suficientes, para depois transferir este conhecimento para aplicações específicas. Este conceito de modelos pré-treinados tem sido muito utilizado nas áreas de NLP e também na área de visão computacional (Bai et al., 2018). Especificamente para NLP, os modelos pré-treinados se aproveitam de bases não rotuladas ou se utilizam delas para criação de bases artificialmente rotuladas, possibilitando dados suficientes para treinamento.

Um modelo de representação pré-treinado, proposto pela equipe de inteligência artificial da empresa Google em 2013, ganhou bastante destaque, batizado como *Word*2Vec, o modelo representa cada palavra como um vetor (*word embedding*) e para geração deste são consideradas as palavras que costumam estar próximas da palavra alvo que o vetor representa, esta abordagem tem o objetivo de dizer quando duas ou mais palavras são similares, como: bonito e lindo, ou criar relações entre palavras, como: homem está para mulher assim como rei está para rainha (MIKOLOV et al., 2013). Para construção deste modelo foram utilizadas 1,6 bilhão de palavras e ele foi considerado na época o estado da arte na representação de texto, possibilitando melhores resultados em diversas atividades de NLP.

Embora tenha bons resultados, o modelo Word2Vec não tem a capacidade de representar o contexto de cada palavra, por exemplo, para a palavra “banco” existe um único vetor que a representa, desta forma nas frases: “Abri uma conta no banco” e “sentei no banco da praça”, a palavra banco tem a mesma representação, podendo causar erros de interpretação nas aplicações de NLP. A dificuldade na geração de modelos pré-treinados contextualizados é causada principalmente pela demora no processamento dos dados, para o modelo Word2Vec o treinamento demorou quase um dia para ser finalizado durante as pesquisas, mesmo utilizando máquinas de última geração do Google.

Com o avanço da tecnologia, principalmente no que tange o poder de processamento, surgiram então os modelos pré-treinados de representação de texto contextualizados, ou seja, para cada palavra são geradas diferentes representações de acordo com seu contexto, desta forma nas frases: “Abri uma conta no banco” e “sentei no banco da praça” são geradas duas representações distintas para palavra “banco”, a primeira referente à instituição financeira e a segunda referente ao objeto feito para se sentar. Estes modelos permitiram avanços em muitas tarefas de NLP, inclusive em tarefas difíceis, como o desafio mantido pela universidade de Stanford nos Estados Unidos, no qual é testada a capacidade de compreensão de leitura através de algoritmos. No desafio é fornecida uma

base de textos e perguntas e se pede para verificar se determinado parágrafo de texto responde determinada pergunta (RAJPURKAR; JIA; LIANG, 2018).

Em novembro de 2018, a empresa Google tornou público o código fonte das técnicas utilizadas para criação de seu novo modelo pré-treinado para NLP, chamado de *Bidirectional Encoder Representantion from Transformer* (BERT). Para construção do modelo pesquisadores desenvolveram uma variedade de técnicas, as quais permitem ao novo modelo de representação de texto utilizar para seu treinamento uma enorme quantidade de dados não rotulados disponíveis na Internet, além de permitir que ele seja utilizado para tarefas gerais de NLP. O modelo constrói uma representação de texto contextualizada, com o diferencial de ser uma contextualização bidirecional, ou seja, ao representar uma palavra é considerado tanto o texto que vem antes dela, como também o texto que vem após ela, conceito difícil de ser implementado até então devido ao grande consumo de processamento requerido (DEVLIN; CHANG, 2018).

O modelo BERT pode ser utilizado de duas formas, a primeira de maneira direta, na qual o modelo pré-treinado é utilizado diretamente para criar uma representação do texto, ou seja, o texto em sua forma natural passa pelo BERT e cria-se sua representação numérica contextualizada (aplicado tanto a palavras como a frases).

A segunda forma de utilização do modelo BERT pode ser feita de maneira indireta, na qual antes de criar uma representação numérica contextualizada para determinado texto, é realizado um ajuste fino no modelo (através de mais treinamento) utilizando os dados do contexto específico ao texto que se quer representar. Nesta segunda forma, o modelo pré-treinado é otimizado para criar a melhor representação para o contexto que se quer trabalhar.

A segunda abordagem, que utiliza o modelo BERT de forma indireta, costuma trazer melhorias substanciais nos resultados obtidos quando comparada a uma abordagem que não utiliza modelos pré-treinados e aproveita-se apenas de dados relacionados ao contexto de uma determinada tarefa (DEVLIN; CHANG, 2018).

Com um modelo pré-treinado de representação textual de alta performance, que consegue gerar uma representação contextualizada otimizada para determinada tarefa, como o BERT se propõe, há a hipótese de que tarefas mais sofisticadas, como detecção de sarcasmo e ironia, podem se tornar possíveis. Desta forma, o objetivo deste trabalho é propor um método para detecção de sarcasmo e ironia, baseado na utilização do modelo pré-treinado de processamento de linguagem natural BERT (Bidirectional Encoder Representation from Transformer), utilizando apenas dados textuais e validar a eficiência do modelo em comparação a outras abordagens de distintas complexidades.

## 2. TRABALHOS RELACIONADOS

Os dois principais trabalhos relacionados ao método proposto são: *A Large Self-Annotated Corpus for Sarcasm*: Khodak, Saunshi e Vodrahalli (2018) e BERT (*Pre-training of Deep Bidirectional Transformers for Language Understanding*): Devlin et al. (2018). O primeiro artigo disponibiliza a base já rotulada e os principais *benchmarks* para o método proposto. O segundo trata-se do modelo pré-treinado que é a base para o método, ambos são descritos a seguir.

*A Large Self-Annotated Corpus for Sarcasm:* Khodak, Saunshi e Vodrahalli (2018) apresentam em seu trabalho uma grande base de dados, construída por eles para pesquisadores de detecção de sarcasmo, e batizam esta base com o nome de *Self-Annotated Reddit Corpus* (SARC). A base conta com mais de 533 milhões de amostras, sendo 1,3 milhão do tipo sarcasmo. A marcação sobre existir ou não sarcasmo na sentença é feita

pelo próprio autor do texto, o que garante a autenticidade da marcação e, por consequência, a qualidade da base.

As amostras que compõem a base foram retiradas da rede social Reddit, na qual usuários se comunicam através de comentários sobre determinado assunto (intitulado com *post*), os comentários se organizam na rede social no formato de árvore, ou seja, todo comentário tem um comentário pai. Os usuários da rede adotaram um método comum para expressão de sarcasmo, eles adicionam o marcador "/s" no final do comentário para indicar que o comentário contém sarcasmo.

Como a aplicação direta da base proposta no artigo é o treinamento e validação de modelos de detecção automática de sarcasmo, os autores constroem uma série de *benchmarks* para a tarefa de classificação dos comentários em sarcástico e não sarcástico. Os *benchmarks* aplicam uma classificação linear sobre três métodos distintos de representação do texto: *Bag-of-Words* (n-gram = 1), *Bag-of-Words* (n-gram = 2), *Sentence embedding* (Glove), mais dois *benchmarks* humano composto por cinco pessoas e um *benchmark* de escolha aleatória, todos aplicados a uma base balanceada e outra desbalanceada.

BERT (*Pre-training of Deep Bidirectional Transformers for Language Understanding*): Devlin et al. (2018) apresentam em seu trabalho um novo modelo de representação de texto pré-treinado. O modelo, batizado como BERT, representa o acrônimo *Bidirectional Encoder Representations from Transformer*. O nome é dado devido ao BERT conseguir criar um modelo de representação de texto de maneira contextualizada, considerando em todas as camadas do modelo palavras à esquerda e à direita do texto que está sendo representado. Pelo nome também fica claro que BERT utiliza a camada *encoder* da arquitetura *Transformer* proposta por Vaswani et al. (2017).

Com o resultado do modelo pré-treinado proposto pelos autores, é possível realizar um ajuste fino com a adição de uma única camada de saída, para então habilitar sua utilização em distintas tarefas da área de NLP. Alguns exemplos das tarefas que podem ser realizadas utilizando o modelo são: classificação de texto, reconhecimento de entidades, perguntas e respostas etc.

A proposta do artigo demostrou resultados superiores aos alcançados até então em diversas tarefas da área de NLP, colocando o modelo no topo de vários desafios. A tarefa *Masked Language Model*, proposta para treinamento, teve um papel muito importante, pois permitiu ao modelo BERT ser bidirecionalmente contextualizado, a tarefa mais comum utilizada até então era unidirecional, a qual solicitava para se predizer uma palavra apenas as suas palavras anteriores ou posteriores.

## 3. METODOLOGIA

Para implementar o método proposto são necessários cinco passos, sendo eles a criação da base de dados, a preparação dos dados, o treinamento do modelo, a aplicação do modelo e a validação do modelo, todos descritos em detalhes abaixo.

**Base de dados:** A base de dados utilizada nesta proposta será a SARC (*Self-Annotated Corpus for Sarcarsm*), proposta por Khodak, Saunshi e Vodrahalli (2018), esta base foi criada especificamente para pesquisadores e tem por finalidade treinar e validar sistemas de detecção de sarcasmo e ironia. A base conta com mais de um milhão de sentenças, dez vezes mais que bases anteriores criadas com o mesmo propósito, as sentenças foram rotuladas pelos próprios autores, excluindo assim um possível viés sobre rótulos feitos por terceiros.

As sentenças presentes na base foram extraídas de um fórum de discussão *on-line* chamado Reddit (https://www.reddit.com/), assim, além da sentença, a base conta também com informações do usuário e assunto.

**Preparação dos dados:** A preparação dos dados para entrada no modelo BERT requer alguns cuidados, como o modelo foi desenhado para ser um modelo genérico de NLP, seu *input* de dados pode ter quatro formatações distintas de acordo com o tipo de tarefa que se quer realizar.

Para a tarefa proposta neste trabalho, classificado pelo BERT como *sentence classification*, a entrada deve ser um arquivo do tipo TSV (dados separados por tabulação) em que cada linha representa uma amostra da base e as colunas devem ter os seguintes valores:

1.Guid: identificador único de cada amostra

2.Label: rotulo da sentença (neste caso 0 ou 1, contém ou não sarcasmo e ironia)

3.Text_b: texto simples da sentença que se vai trabalhar, coluna opcional, deve ser utilizado apenas para tarefas que envolvem pares de sentenças (não é o caso deste trabalho, no qual a mesma será desconsiderada e, por isso, preenchida apenas com o caractere ‘a’)

4.Text_a: texto simples da primeira sentença que se vai trabalhar (sentença a analisar se contém sarcasmo e ironia)

A Figura 1 apresenta um exemplo real utilizado para o treinamento.

Figura 1 – Formato de entrada

```
23  1   a   i am deeply shocked and surprised at this development
24  1   a   this is pure xenophobia against vaticanians!
25  1   a   but the terrorists!
26  0   a   india is going to go ape-shit
27  1   a   ha ha ha ha its funny.
28  1   a   comic sans is such a badass font.
29  1   a   fake - i've been getting this chain email since 1999.
30  0   a   but this is the *best health care in the world*.
31  0   a   but steve didn't tell me... is it magical and easy to use?
32  0   a   lying is the norm today, truth is the exception.
33  1   a   """wow... profound"""
```

Fonte: Elaborado pelo autor.

Os dados disponíveis no SARC estão separados em três arquivos, o primeiro é um arquivo do tipo JSON, nomeado ‘comments.json’, que contém todos os comentários disponíveis na base, assim como informações adicionais, como identificador do autor, data de criação, o comentário em si e outros, como é possível ver na Figura 2. Todos os comentários disponíveis têm um identificador único que permite acessá-lo.

Figura 2 – Comentários Reddit


```
{'author': 'monkeyman114',
 'created_utc': 1234057973,
 'date': '2009-02',
 'downs': 0,
 'score': 1,
 'subreddit': 'reddit.com',
 'text': 'I think the government should track every mormon in the country for subversive activity.',
 'ups': 1}
```


Fonte: Elaborado pelo autor.

Os outros dois arquivos contêm referências para os dados de treino e teste (train_balanced.csv e test_balanced.csv), estão em formato CSV e contêm as seguintes informações separados pelo caractere pipe '|' e por espaço ' ':

1.Identificador do tópico postado (pipe)
2.Identificador do comentário 1 (espaço)
3.Identificador do comentário 2 (pipe)
4.Label comentário 1 (espaço)
5.Label comentário 2

A Figura 3 ilustra os arquivos com as referências para treino e teste da base no formato descrito.

Figura 3 – Base com referências

| |
|---|
| 8139p\|c07yhlm c07yoiw\|1 0 |
| bnmod\|c0nnujf c0nofrs\|1 0 |
| boess\|c0ns0me c0nru3a\|1 0 |
| bppwt\|c0nyfx0 c0nylfu\|1 0 |
| bqs92\|c0o4fkr c0o4uu3\|1 0 |
| brelu\|c0o78ue c0o77tt\|0 1 |
| bsabi\|c0obdpq c0obx5c\|0 1 |
| btr9t\|c0ojspf c0oipwz\|1 0 |
| bwfbk\|c0oxmjp c0owyod\|0 1 |
| hoxvh\|c1x6nos c1x6e26\|0 1 |
| hqa1x\|c1xiujs c1xj4e2\|1 0 |
| hpopn\|c1xeuca c1xdepf\|0 1 |
| httvg\|c1yhlfy c1ydyi4\|0 1 |

Fonte: Elaborado pelo autor.

Com estes arquivos de treino e testes disponíveis é necessário realizar a busca de cada comentário no arquivo comments.json e, assim, montar a entrada no formato esperado pelo modelo BERT conforme Figura 3.

A base de treino é composta por 257.081 amostras e a base de teste é composta por 64.665 amostras. É importante que as duas não se misturem em momento algum para que a comparação dos resultados seja imparcial.

**Treinamento do modelo:** De forma macro, as amostras da base SARC, depois de pré-processadas, passam pelas doze camadas *encoder*, após este passo é utilizada a representação do *token* especial [CLS], que alimenta a camada adicional de classificação e informa a probabilidade do texto conter ou não conter sarcasmo e ironia.

A Figura 4 mostra este macro fluxo e seus quatro grandes estágios, os quais são detalhados a seguir:

Estágio 1: A entrada de sua linguagem natural é convertida Ids (*tokens*) por palavra (vocabulário BERT, neste trabalho é utilizada a língua inglesa), é realizado o processo de *wordpiece*, são adicionados ou removidos *tokens* para que toda entrada respeite o tamanho de 128 e então são inseridos os *tokens* especiais, como [CLS] e [SEP], e, então, é alimentado o modelo.

Estágio 2: Os dados de entrada passam por doze camadas *encoder*. Dentro de cada *Encoder* o dado passa pela camada de *Attention*, na qual é gerada uma nova representação

para cada *token*, de acordo com os demais *tokens* da entrada, e em seguida por uma camada FFNN, gerando a representação final de cada palavra para esta camada *encoder*.

Este processo se repete para os doze *Encoders*, tendo como entrada de cada *Encoder* a saída do anterior. Cada *token* de entrada tem seu próprio caminho durante todo o treinamento até que o último *Encoder* gere a saída para cada *token* de entrada.

Estágio 3: No último *Encoder* há a representação final de cada *token* de entrada, para se obter esta representação são utilizados os estados da última camada oculta, resultando em um vetor de tamanho 768 (tamanho da camada FFNN) para cada entrada.

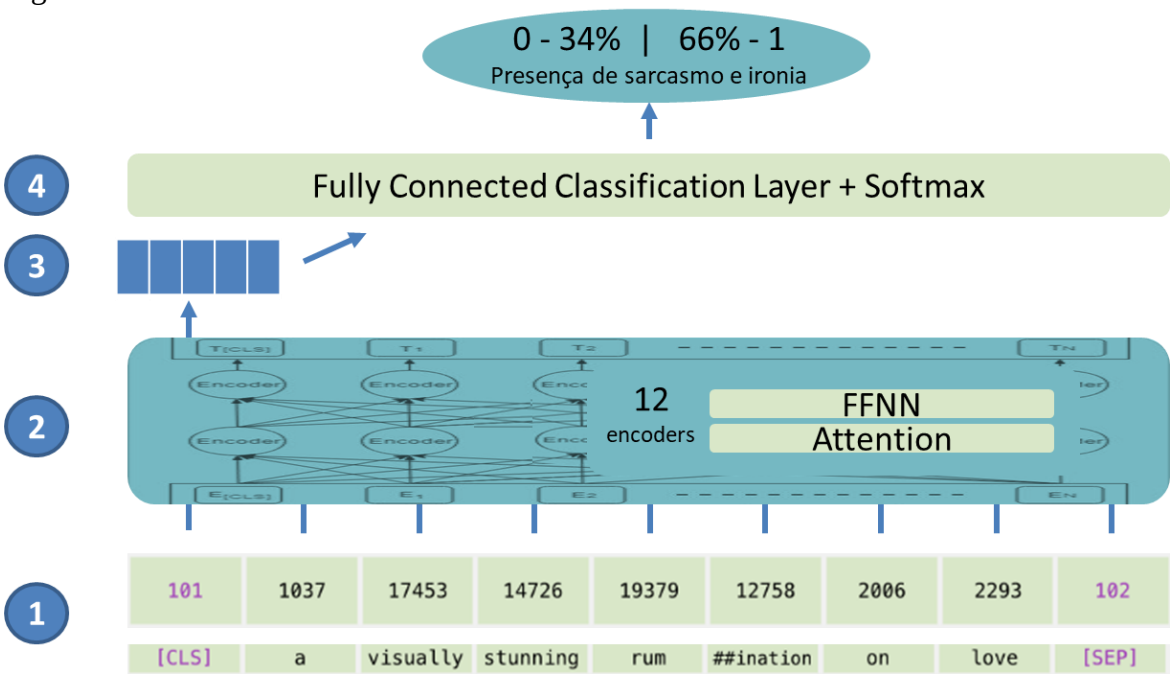
Como neste trabalho é analisada uma sentença por completo e não cada palavra utiliza-se o *token* especial [CLS], este *token* foi designado pela arquitetura BERT para representar a sentença toda. Em seu treinamento, mais especificamente na tarefa NSP, o BERT se especializa em conseguir representar uma sentença por completo. Desta forma, é passado para o próximo estágio a representação do *token* [CLS].

Um ponto importante a se reforçar é que para tornar este treinamento viável, todos os parâmetros são inicializados com os parâmetros do modelo pré-treinado BERT Base.

Estágio 4: A representação da sentença é enviada para uma camada totalmente conectada que tem tamanho variado de acordo com o número de classes existentes ao problema que se quer resolver. Como no problema existem apenas duas classes (contém e não contém ironia e sarcasmo), é gerado um vetor de duas posições, cada uma representa uma das classes possíveis e seu valor é a chance de que esta classe seja verdadeira. Após isso, a camada *Softmax* transforma esta chance em probabilidade, na qual a soma de todas as probabilidades resulta um, e, então, a posição de maior probabilidade é escolhida, produzindo a saída do classificador.

Todo este processo é treinado através do processo padrão de *backpropagation* e o modelo é otimizado para maximizar a assertividade sobre o problema de detecção de sarcasmo e ironia.

Figura 4 – Macro fluxo de treinamento




Fonte: Elaborado pelo autor.

**Aplicação do modelo:** A aplicação do modelo treinado para realizar as predições sobre o texto de entrada segue exatamente a mesma linha do treinamento, com a exceção de que na predição não é realizado nenhum novo aprendizado pela rede, ou seja, os parâmetros (pesos) da rede ficam congelados.

Para cada linha de texto de entrada é gerada uma saída com dois valores, sendo estes valores a probabilidade do texto não conter sarcasmo e ironia e a probabilidade do texto conter sarcasmo e ironia.

**Avaliação do método:** Para construir a base SARC em sua versão balanceada (versão utilizada neste trabalho), Khodak, Saunshi e Vodrahalli (2018), os autores da proposta, seguiram a seguinte metodologia: para cada postagem na rede social Reddit, que contém ao menos um comentário marcado como sarcástico pelo próprio autor, se adicionou na base um comentário sarcástico e um não sarcástico desta postagem.

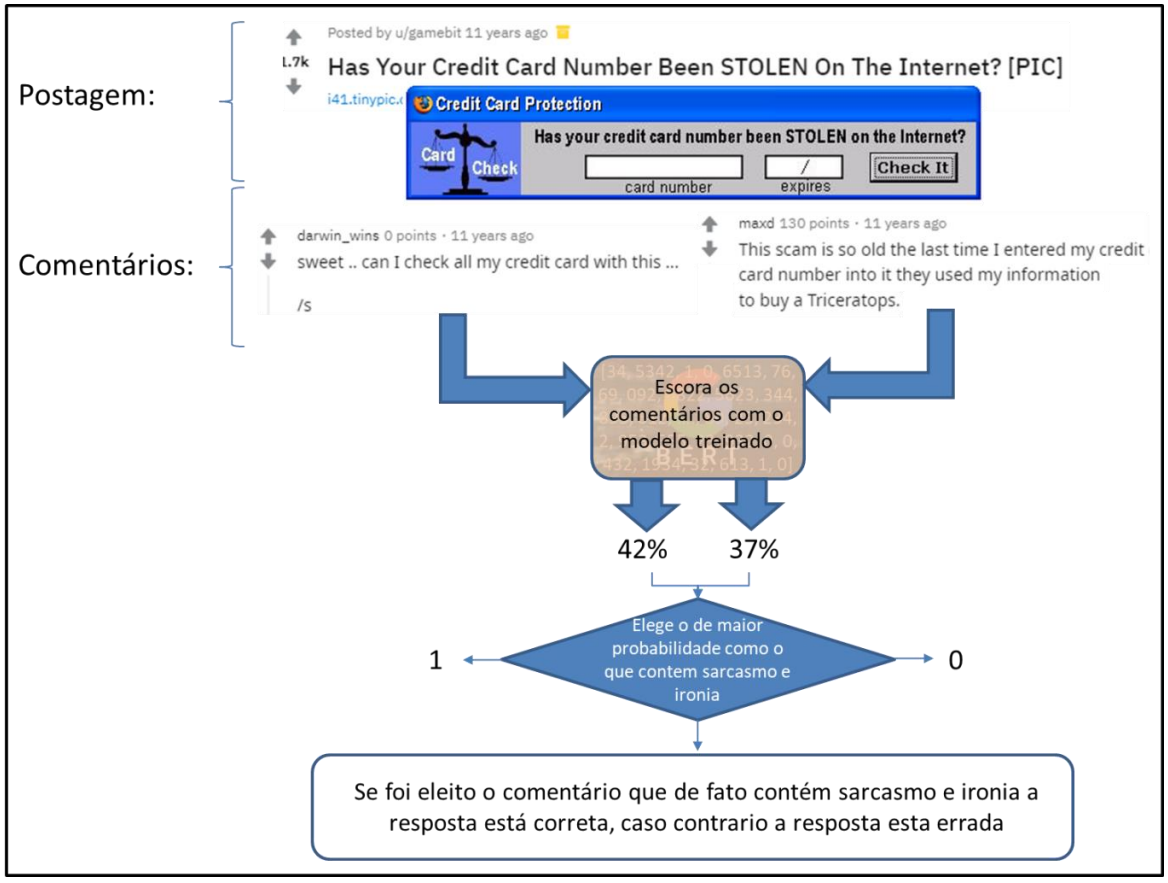
Toda esta estrutura que relaciona qual comentário está relacionado à qual postagem está presente na estrutura da base SARC, através de seu arquivo JSON, que contém detalhes das informações de cada comentário (postagem origem, autor, data, entre outros). Desta forma, por mais que o modelo treinado tenha como objetivo predizer se um determinado comentário contém sarcasmo e ironia, a avaliação proposta pelos autores é um pouco mais elaborada.

Os autores da base SARC propõem avaliar o modelo verificando o seguinte desafio: dados dois comentários realizados na mesma postagem, qual dos dois é o comentário que contém sarcasmo e ironia?

Nos modelos criados como *benchmark* no artigo em que se propõe a base SARC, os autores utilizaram como metodologia para definir qual dos comentários é o que contém sarcasmo e ironia o seguinte fluxo: passam-se os dois comentários por um mesmo classificador, é verificada a probabilidade de cada um deles conter sarcasmo e ironia, em seguida as probabilidades são comparadas e é eleito como o que contém sarcasmo e ironia o de maior probabilidade.

Uma vez eleito qual comentário de cada postagem contém sarcasmo e ironia, haverá uma resposta por postagem sinalizando se foi marcado o comentário correto. A avaliação então é realizada verificando a acurácia sobre o resultado da marcação de cada postagem. Todo o fluxo de avaliação pode ser observado na Figura 5.

Figura 5 – Fluxo de avaliação



Fonte: Elaborado pelo autor.

## 4. RESULTADOS

No treinamento e teste deste trabalho estão sendo utilizadas as mesmas amostras para garantir uma comparação imparcial. A fim de enriquecer a comparação foram desenvolvidos pelo autor deste trabalho mais dois modelos *baseline*, totalizando nove modelos, cinco desenvolvidos por Khodak, Saunshi e Vodrahalli (2018), um desenvolvido por Hazarika et al. (2018) e três desenvolvidos pelo autor. A seguir segue o detalhe de cada *baseline*:

*Sentence Embedding*: criado por Khodak, Saunshi e Vodrahalli (2018), o modelo utiliza o modelo pré-treinado GLOVE (*Global vectors for word representation*), desenvolvido por Pennington, Socher e Manning (2014) para representar cada comentário. Em seguida é aplicada uma regressão logística para classificar se contém ou não sarcasmo e ironia.

*Bag of Word*s (1-gram): criado por Khodak, Saunshi e Vodrahalli (2018), o modelo utiliza a abordagem *Bag of Word*s, comentada na introdução, e considera apenas *tokens* únicos para criação do vocabulário e geração da representação dos comentários. Em seguida é aplicada uma regressão logística para classificar se contém ou não sarcasmo e ironia.

*Bag of Words* (2-gram): criado por Khodak, Saunshi e Vodrahalli (2018), o modelo utiliza a abordagem *Bag of Words*, e considera *tokens* únicos e também a combinação de dois *tokens* que aparecem em sequência para criação do vocabulário e geração da representação dos comentários. Em seguida é aplicada uma regressão logística para classificar se contém ou não sarcasmo e ironia.

Humano (individual): criada por Khodak, Saunshi e Vodrahalli (2018), esta abordagem contou com cinco humanos realizando o mesmo desafio que os algoritmos, ou seja, dado dois comentários classificar qual contém e qual não contém ironia e sarcasmo. Para se chegar a acurácia final considera-se a média de acertos dos cinco humanos.

Humano (grupo): criada por Khodak, Saunshi e Vodrahalli (2018), esta abordagem também utilizou cinco humanos, porém ao invés de considerar o resultado final sendo a média do resultado de todos, se construiu um classificador único que considera para classificação de cada comentário a classe mais votada pelos cinco humanos.

CASCADE: criada por Hazarika et al. (2018), esta abordagem utiliza tanto o conteúdo do texto como também a personalidade do autor, que é extraída através do conteúdo do texto por um modelo adicional.

*Word*2Vec: criada pelo autor deste trabalho, esta abordagem utiliza o modelo pré-treinado *Word*2Vec do Google para extrair a representação vetorial de cada palavra, em seguida é realizada a média destes vetores para criar a representação da sentença e, por fim, é aplicada uma regressão logística para classificar se o comentário contém ou não sarcasmo e ironia.

BERT extrator de características: criada pelo autor deste trabalho, esta abordagem utiliza o modelo pré-treinado BERT para criar a representação de cada comentário. Com a representação extraída é treinado uma FFNN com função de ativação *softmax* para classificar se o comentário contém ou não sarcasmo e ironia.

BERT otimizado para SARC: criada pelo autor deste trabalho, esta abordagem, explicada em detalhes no item três deste trabalho, utiliza a arquitetura proposta por Devlin (2018) mais os parâmetros aprendidos em seu modelo pré-treinado para otimizar um modelo de detecção de sarcasmo e ironia, utilizando dados da base SARC.

Na Tabela 1 observa-se a acurácia de todos os métodos propostos explicados acima, ordenados da menor para a maior acurácia, com isso é possível analisar as capacidades de cada um para o desafio de detectar sarcasmo e ironia.

Tabela 1 - Acurácia dos métodos propostos

| Autor | Método | Base | Acuracia |
|---|---|---|---|
| Elaborado pelo autor | BERT Extrator de caracteristicas | SARC | 63.0 |
| Elaborado pelo autor | Word2vec | SARC | 63.7 |
| Khodak, Saunshi e Vodrahalli (2018) | Glove | SARC | 71.0 |
| Khodak, Saunshi e Vodrahalli (2018) | BoW (1-gram) | SARC | 73.2 |
| Khodak, Saunshi e Vodrahalli (2018) | BoW (2-gram) | SARC | 75.8 |
| Hazarika *et al.* (2018) | CASCADE | SARC | 78.0 |
| Elaborado pelo autor | BERT otimizado para SARC | SARC | 78.7 |
| Khodak, Saunshi e Vodrahalli (2018) | Human (average) | SARC | 81.6 |
| Khodak, Saunshi e Vodrahalli (2018) | Human (major) | SARC | 92.0 |

Fonte: Elaborado pelo autor.

Os resultados observados na Tabela 1 podem parecer um tanto quanto surpreendentes, não apenas positivamente para o caso do modelo principal proposto por este trabalho, mas também por modelos que obtiveram performance pior que abordagens simples. A seguir são observados caso a caso.

O modelo "BERT como extrator de características" obteve o pior resultado (acurácia de 63%), mesmo representando os comentários como o modelo pré-treinado puro, no qual se esperava criar uma boa representação que contribuísse com o treinamento do algoritmo, isto não se refletiu no resultado. Uma hipótese para que um modelo pré-

treinado puro não funcione pode ser a diferença de vocabulário utilizado para treinamento de modelo contra o vocabulário da tarefa que se está tentando resolver.

O modelo criado a partir do *Word*2Vec obteve um resultado (acurácia de 63,7%) melhor que BERT como extrator de características, porém ainda está longe dos demais modelos. Aqui vale ressaltar uma dificuldade do modelo *Word*2Vec para representação de sentenças, por concepção o modelo *Word*2Vec inicialmente foi criado para representar apenas palavras, para se trabalhar com sentenças é necessária uma adaptação de seu resultado, com a finalidade de se representar a sentença, para este *baseline* foi utilizado uma adaptação simples, que pode ter contribuído para o mau desempenho.

O modelo GLOVE obteve um resultado (acurácia de 71%) abaixo da mediana dos demais métodos, porém melhor que o modelo *Word*2Vec. Até aqui há uma surpresa negativa com os modelos de *word embedding* obtendo resultados piores que modelos mais simples que serão vistos a seguir.

O modelo *Bag of Words* (1-gram), surpreendentemente obteve resultado (acurácia de 73,2%) melhor do que os modelos de *word embedding*. Aqui a hipótese é que a base SARC tenha um vocabulário muito específico que não se adapta bem a modelos pré-treinados, outra hipótese pode ser de que os comentários por possuírem tamanhos pequenos (maioria entre sete e quinze palavras), permitem que o *Bag of Words* consiga criar uma boa representação.

O modelo *Bag of Words* (2-gram) obteve ótimo resultado (acurácia de 75,8%), seguindo a performance do *Bag of Words* (1-gram) bateu os modelos básicos de *word embedding* e o próprio *Bag of Words* (1-gram), dado que a representação 2-gram pode capturar mais características presentes nos comentários.

O modelo CASCADE, proposto por Hazarika et al. (2018), apresentou ótimo resultado (acurácia de 78,0%) conforme esperado, dado que o modelo explora mais variáveis e modelos adicionais que inferem a personalidade do autor do texto.

O modelo "BERT otimizado para SARC", método proposto neste trabalho, explicado em detalhes no capítulo 3, obteve o melhor resultado (acurácia de 78,7%) comparado aos demais modelos de aprendizado de máquina. Este modelo obteve performance aproximadamente 50% melhor que o modelo que utilizou BERT puro, e superou os modelos de *Bag of Words* e CASCADE (embora comparado ao CASCADE o resultado seja muito próximo, 0,7% de diferença). Este modelo perdeu apenas para os *baselines* realizados por humanos.

Na Tabela 2 pode-se observar também a acurácia do modelo "BERT otimizado para SARC" em sua base de treino e sua base de teste (85% e 70% respectivamente). Os resultados mostram um relativo equilíbrio entre treino e teste, indicando que o modelo não contém demasiado *overfitting*, mesmo sendo passível de melhora. Infelizmente nem todos os modelos de comparação disponibilizam estes números para que seja possível avaliar em maior profundidade.

Tabela 2 – Performance Treino e Teste

| Modelo | Acurácia | | |
|---|---|---|---|
| | Treino | Teste | Experimento |
| BERT otimizado para SARC | 85% | 70% | 78,7% |

Fonte: Elaborado pelo autor.

Os modelos Humano (individual) e Humano (grupo) obtiveram os melhores resultados (acurácias de 81,6% e 92,0%, respectivamente), aqui é interessante observar a

diferença no resultado dos modelos. Primeiro, a diferença do modelo Humano (individual) para os modelos de aprendizado de máquina (2,9% de diferença para o melhor modelo "BERT otimizado para SARC"), trata-se de uma diferença não tão grande. Já a diferença do modelo humano (grupo) para os modelos de aprendizado de máquina e para o modelo Humano (individual) mostram grandes diferenças (13,3% e 10,4% respectivamente). A hipótese aqui é que são necessários conhecimentos distintos para poder identificar sarcasmo e ironia, por isso um grupo de humanos tem acerto melhor de que humanos analisando de forma individual.

## 5. CONCLUSÕES

Neste artigo foi proposto um método com base na arquitetura do modelo de NLP pré-treinado BERT, disponibilizado pelo Google, com a adição de uma camada FFNN com ativação *Softmax*, treinada de forma supervisionada com o objetivo de detectar sarcasmo e ironia em texto, utilizando a base de dados SARC disponibilizada por Khodak, Saunshi e Vodrahalli (2018).

O problema de detecção de sarcasmo e ironia em texto conta com os desafios já conhecidos no campo de NLP, sendo o principal deles representar corretamente o contexto expressado no texto. Este contexto se torna mais difícil de representar e interpretar à medida que o campo de observação se torna maior, ou seja, representar uma palavra é difícil e a dificuldade aumenta ao se representar uma frase, um parágrafo, e assim por diante, isso se dá ao fato de ser difícil criar relações e construir um contexto.

Além dos desafios já conhecidos no campo de NLP, o problema de detecção de sarcasmo e ironia em texto conta com seus próprios desafios, tendo também como necessidade conhecer o contexto sobre o tema que se está analisando. Este desafio é potencializado com as dificuldades gerais de NLP (minimizadas pelo modelo BERT), na qual o contexto é difícil de ser bem representado e torna o problema complexo de se resolver.

Como para o problema de detecção de sarcasmo e ironia através da base SARC não existe mais contexto fora o que está sendo escrito em cada comentário, o desafio tratado de forma tradicional com o uso de aprendizado de máquina e NLP se resumiria em separar as palavras que melhor distinguem um comentário de conter ou não sarcasmo e ironia. Porém com o uso do modelo BERT cada palavra pode ter uma representação distinta de acordo com seu contexto, então é possível ter uma maior diferenciação entre os comentários que contém ou não sarcasmo e ironia.

O treinamento do método proposto é inicializado com os parâmetros do modelo pré-treinado BERT (aqui há a transferência de conhecimento de um modelo para o outro), o que permite uma convergência de treino em um tempo muito menor (porém ainda grande, foram consumidas 30 horas, devido ao grande número de amostras presentes na base), pois as camadas iniciais do modelo tendem a se manter estáveis, pois têm um conhecimento mais genérico sobre o texto (já aprendido quando treinado pelo Google).

As camadas mais profundas e a nova camada especializada tendem a sofrer maiores alterações em seus pesos, pois aqui se cria o conhecimento específico sobre o problema que se quer resolver, no caso deste trabalho, a detecção de sarcasmo e ironia. Desta forma, tem-se o modelo treinado proposto neste trabalho.

O resultado do método proposto neste trabalho (Acurácia de 78,7%) atingiu as expectativas superando em performance os demais *benchmarks* que utilizaram aprendizado de máquina, só não foi possível atingir resultados melhores que os *benchmarks* realizados por humanos classificando individualmente ou em grupos (acurácias de 81,6% e 92% respectivamente).

Porém, quando comparado o resultado do método proposto apenas com o resultado do modelo humano individual, não há uma diferença muito grande (apenas 2,9%), mostrando que a máquina está próxima a superar o resultado de um humano sozinho na realização da tarefa de detecção de sarcasmo e ironia, certamente neste contexto controlado da base SARC (mesmo a base SARC representando comentários reais de uma rede social).

Outro fator interessante foi analisar a grande diferença entre os resultados dos modelos Humano Individual e Humano Grupo (10,4%), aqui a hipótese é de que com mais humanos como é o caso do modelo Humano Grupo se tem maior conhecimento e contexto sobre diversos temas, ou seja, um humano pode saber mais sobre o tema "esportes", outro sobre "política" e assim por diante, criando uma base de conhecimento muito maior que um humano sozinho. A mesma comparação pode ser feita para com o método proposto neste trabalho, pois o método proposto só tem conhecimento sobre uma determinada tarefa, faltando maior conhecimento e contexto sobre outros temas.

Sobre o experimento do modelo BERT (puro) aplicado diretamente à detecção de sarcasmo e ironia esperava-se um resultado melhor. Dado que o modelo foi criado com a proposta de ser genérico e de apresentar melhores resultados em todas as tarefas de NLP, tinha-se a expectativa de que se alcançasse ao menos resultados próximos aos modelos de *word embedding* como o *Word*2Vec e o GLOVE.

Outro ponto importante que se pode concluir é de que os modelos de *word embedding* (BERT puro, Word2Vec e Glove), mesmo dispondo de técnicas mais sofisticadas e dados externos em seu treinamento não conseguem resultados melhores do que os modelos de técnica simples (*Bag of Words*) que utilizam dados do problema que se quer resolver. O mesmo não se aplica ao método proposto neste trabalho, dado que são utilizados dados externos para construção do modelo BERT pré-treinado, porém posteriormente ele sofre um ajuste para o problema de detecção de sarcasmo e ironia utilizando os dados da base SARC rotulada e desta forma obteve obtém melhores resultados.

Por fim, é possível concluir que o objetivo proposto neste trabalho foi alcançado, foi criado um método baseado no modelo BERT para detecção de sarcasmo e ironia utilizando apenas os dados textuais. Os resultados alcançados pelo método proposto foram satisfatórios, superando outras metodologias de aprendizado de máquina e se aproximando de humanos realizando a mesma tarefa, mesmo trabalhando sem o contexto do comentário que se vai classificar.

## 6. RECOMENDAÇÕES

Com o objetivo de evoluir ainda mais no tema de pesquisa, são sugeridos os seguintes trabalhos futuros:

Realizar estudo comparativo entre o método proposto e a utilização do BERT puro para identificar os fatores que levaram os dois a terem grande diferença em seus resultados.

Treinar um modelo com a arquitetura proposta em uma base de sarcasmo e ironia no idioma português.

Aplicar o método proposto e treinado neste trabalho em outra base no idioma inglês e verificar a capacidade de identificar sarcasmo e ironia mesmo em outras situações que não a de um fórum de discussões.

## REFERÊNCIAS


ADHIKARI, A. et al. Docbert: BERT for document classification. 2019. Disponível em: http://arxiv.org/abs/1904.08398. Acesso em: 20 abr. 2019.

ALAMMAR, J. The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning). 2018. Disponível em: http://jalammar.github.io/illustrated-bert/. Acesso em: 20 abr. 2019.

ALAMMAR, J. The Illustrated Transformer. 2018. Disponível em: https://jalammar.github.io/illustrated-transformer/. Acesso em: 20 abr. 2019.

ATTARDO, S. Irony as relevant inappropriateness. Journal of Pragmatics, Youngstown, Estados Unidos, v. 32, p. 793-826, 2000.

BA, J.; KIROS, J.; HINTON, G. Layer Normalization. 2016. Disponível em: https://arxiv.org/abs/1607.06450. Acesso em: 15 jun. 2019.

BAI, M. et al. Transfer pretrained sentence encoder to sentiment classification. In: 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC), Guangzhou, China, 2018. p. 423–427.

BATIMARCHI, S. A diferença entre dados estruturados e não estruturados. 2015. Disponível em: http://docmanagement.com.br/03/06/2015/a-diferenca-entre-dados-estruturados-e-nao-estruturados/. Acesso em: 03 mar. 2019.

BROWNLEE, J. A Gentle Introduction to the Bag-of-*Word*s Model. 2017. Disponível em: https://https://machinelearningmastery.com/gentle-introduction-bag-*word*s-model. Acesso em: 03 mar. 2019.

CHENG, J.; DONG, L.; LAPATA, M. Long short-term memory-networks for machine reading. In: Conference on Empirical Methods in Natural Language Processing (EMNLP), Austin, EUA, 2016.

CHIMENTI, M. Are deep neural nets “Software 2.0”?. 2017. Disponível em: https://www.michaelchimenti.com/2017/11/deep-neural-nets-software-2-0/. Acesso em: 16 mar. 2019.

CHOLLET, F. Xception: Deep Learning with Depthwise Separable Convolutions. 2016. Disponível em: https://arxiv.org/abs/1610.02357. Acesso em: 11 maio 2019.

Deng, J. et al. ImageNet: A Large-Scale Hierarchical Image Database. In: Proceedings of the 2009 IEEE conference on computer vision and pattern recognition (CVPR), Miami, EUA, 2009, p. 248–255.

DEVLIN, J. et al. BERT: pre-training of deep bidirectional transformers for language understanding. 2018. Disponível em: http://arxiv.org/abs/1810.04805. Acesso em: 30 mar. 2019.

DEVLIN, J.; CHANG, M-W. Open Sourcing BERT: State-of-the-Art Pre-training for Natural Language Processing. 2018. Disponível em: https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html. Acesso em: 30 mar. 2019.

GEHRING, J. et al., Convolutional Sequence to Sequence Learning. 2017. Disponível em: http://arxiv.org/abs/1705.03122. Acesso em: 11 maio 2019.

GOLDBERG, Y.; HIRST, G. Neural Network Methods in Natural Language Processing. Morgan & Claypool Publishers, 2017.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. Deep Learning. [S.l.]: MIT Press, 2016. Disponível em: http://www.deeplearningbook.org. Acesso em: 30 abr. 2019.

GRAVES, A. Sequence Transduction with Recurrent Neural Networks. 2012. Disponível em: http://arxiv.org/abs/1211.3711. Acesso em: 13 abr. 2019.

GREFENSTETTE, E. et al. Learning to Transduce with Unbounded Memory. 2015. Disponível em: http://arxiv.org/abs/1506.02516. Acesso em: 12 abr. 2019.

HAZARIKA et al. CASCADE: Contextual Sarcasm Detection in Online Discussion Forums. 2018. Disponível em: https://arxiv.org/abs/1805.06413. Acesso em: 16 jun. 2019.

HE, K. et al. Deep residual learning for image recognition. In: Proceedings of the 2016 IEEE conference on computer vision and pattern recognition (CVPR), Las Vegas, EUA, 2016, p. 770–778.

IOFFE, S.; SZEGEDY, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. 2015. Disponível em: http://arxiv.org/abs/1502.03167. Acesso em: 15 jun. 2019.

IRONIA. In: DICIONÁRIO online do Michaelis. Disponível em: https://michaelis.uol.com.br/. Acesso em 06 abr. 2019.

JOSHI, A.; BHATTACHARYYA, P.; CARMAN, M. Automatic Sarcasm Detection: A Survey. 2016. Disponível em: https://arxiv.org/abs/1602.03426. Acesso em: 23 mar. 2019.

JOSHI, M. et al. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL), Vancouver, Canadá, 2017, p. 1601–1611.

KHODAK, M.; SAUNSHI, N.; VODRAHALLI, K. A large self-annotated corpus for sarcasm. 2017. Disponível em: http://arxiv.org/abs/1704.05579. Acesso em: 16 jun. 2019.

LING, J.; KLINGER, R. An Empirical, Quantitative Analysis of the Differences Between Sarcasm and Irony. In: European Semantic Web Conference (ESWC), 2016.

LIU, B. Sentiment analysis and subjectivity. Handbook of Natural Language Processing. Boca Raton, EUA: CRC Press, Taylor and Francis Group, 2010.

MACHINE LEARNING CRASH COURSE. Embeddings: Translating to a Lower-Dimensional Space. Disponível em: https://developers.google.com/machine-learning/crash-course/embeddings/translating-to-a-lower-dimensional-space. Acesso em: 9 mar. 2019.

MEDIUM. Seq2seq pay Attention to Self Attention: Part 1. 2018. Disponível em: https://medium.com/@bgg/seq2seq-pay-attention-to-self-attention-part-1-d332e85e9aad/. Acesso em: 25 jun. 2019.

MEDIUM. Seq2seq pay Attention to Self Attention: Part 2. 2018. Disponível em: https://medium.com/@bgg/seq2seq-pay-attention-to-self-attention-part-2-cf81bf32c73d/. Acesso em: 25 jun. 2019.

MENABREA, L. F. Sketch of the Analytical Engine invented by Charles Babbage. 1843. Disponível em: https://books.google.com.br/books?id=hPRmnQEACAAJ. Acesso em: 02 mar. 2019.

MIKOLOV, T. et al. Efficient Estimation of *Word* Representations in Vector Space. 2013. Disponível em: http://arxiv.org/abs/1301.3781. Acesso em: 30 abr. 2019.

NWANKPA, C. et al. Activation Functions: Comparison of Trends in Practice and Research for Deep Learning. 2018. Disponível em: https://arxiv.org/abs/1811.03378. Acesso em: 03 mar. 2019.

OLIVEIRA, M. R. G. Aplicação do Modelo Bidirectional Encoder Representation from Transformer (BERT) para Detecção de Ironia e Sarcasmo em Texto. São Paulo, 2020. 108 f. Dissertação (Mestrado Profissional em Engenharia de Software) - Coordenadoria de Ensino Tecnológico, Instituto de Pesquisas Tecnológicas do Estado de São Paulo, São Paulo, 2020.

PAPINENI, K. et al. BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, EUA, 2002. p. 311–318.

RAJPURKAR et al. Squad: 100,000+ questions for machine comprehension of text. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP), Austin, EUA, 2016, p. 2383–2392.

RAJPURKAR, P.; JIA, R.; LIANG, P. Know what you don't know: Unanswerable questions for squad. 2018. Disponível em: http://arxiv.org/abs/1806.03822. Acesso em: 13 abr. 2019.

RILOFF, E. et al. Sarcasm as contrast between a positive sentiment and negative situation. In: Conference on Empirical Methods in Natural Language Processing (EMNLP), Salt Lake City, EUA, 2013.

SAMMUT, C.; WEBB, G. I. Encyclopedia of Machine Learning. Boston, EUA. p. 986–987. 2017.

SARKAR, D. Text Analytics with Python: A Practical Real-World Approach to Gaining Actionable Insights from Your Data. Apress. 2016.

SARKAR, D.; BALI, R.; GHOSH, T. Hands-On Transfer Learning with Python: Implement advanced deep learning and neural network models using TensorFlow and Keras. Packt Publishing. 2018. 440p.

SCHMIDHUBER, J. Deep learning in neural networks: An overview. Neural Networks, v. 61, p. 85-117, 2015.

SCHUSTER, M.; NAKAJIMA K. Japanese and Korean voice search. In: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Kyoto, Japão, 2012.

TABACARU, S. Uma visão geral das teorias do humor: aplicação da incongruência e da superioridade ao sarcasmo. Revista Eletrônica de Estudos Integrados em Discurso e Argumentação, p. 115-136, 2015.

TAN, C. et al. A survey on deep transfer learning. 2018. Disponível em: http://arxiv.org/abs/1808.01974. Acesso em: 12 abr. 2019.

TAYLOR, W. L. Cloze procedure: a new tool for measuring readability. Journalism Quarterly, v. 30, p. 415–433, 1953.

TEPPERMAN, J.; TRAUM, D; NARAYANAN, S. "Yeah right": sarcasm recognition for spoken dialogue systems. In: INTERSPEECH 2006 Ninth International Conference on Spoken Language Processing (ICSLP), Pittsburgh, EUA, 2006.

THE ALLEN INSTITUTE FOR ARTIFICIAL INTELLIGENCE. Leaderboard – SWAG. Disponível em: https://leaderboard.allenai.org/swag/submissions/public. Acesso em: 26 set. 2019.

THE GENERAL LANGUAGE UNDERSTANDING EVALUATION – GLUE. Leaderboard. Disponível em: https://gluebenchmark.com/leaderboard. Acesso em: 26 set. 2019.

THE STANFORD QUESTION ANSWERING DATASET – SQUAD. Leaderboard. Disponível em: https://rajpurkar.github.io/SQuAD-explorer/. Acesso em: 26 set. 2019.

TOWARDS DATA SCIENCE. Coding Neural Network. 2018. Disponível em: https://towardsdatascience.com/coding-neural-network-forward-propagation-and-backpropagtion-ccf8cf369f76. Acesso em: 16 mar. 2019.

UC BUSINESS ANALYTICS R PROGRAMMING GUIDE. Creating text features with bag-of-*word*s, n-grams, parts-of-speach and more. 2018. Disponível em: http://uc-r.github.io/creating-text-features. Acesso em: 9 mar. 2019.

VASWANI, A. et al. Attention is all you need. 2017. Disponível em: http://arxiv.org/abs/1706.03762. Acesso em: 27 abr. 2019.

WANG, A. et al. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. 2019. Disponível em: http://arxiv.org/abs/1905.00537. Acesso em: 28 set. 2019.

ZELL, A. Simulation Neuronaler Netze, 1995.

ZELLERS, R. et al. Swag: A large-scale adversarial dataset for grounded commonsense inference. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP), Bruxelas, Bélgica, 2018.