

Aprendizado de Máquina

DETECCÃO DESPAM EM EMAILS

GIL DE ALMEIDA, PEDRO BRUCE E
THIAGO CASTILHO

AGENDA

1. O QUE SÃO SPAMS?
2. VETORIZAÇÃO
3. PRÉ-PROCESSAMENTO
4. NAIVE BAYES
5. SVM
6. KNN
7. BERT
8. IMPLEMENTAÇÃO E RESULTADOS



O QUE SÃO SPAMS?



Um *spam* é, sobretudo, um *e-mail* não desejado, que pode ser enviado em larga escala.



A grande maioria das mensagens de *spam* são de natureza comercial, como divulgação, mas o maior perigo está naqueles que contém *links* maldosos ou *malwares* como anexo.



Os *spammers* coletam os endereços de *e-mail* em salas de bate-papo, listas, sites, e muitas vezes são comprados de empresas ou de outros spammers.



O envio de *spams* tem crescido constantemente desde o início dos anos 90. Em 2014, foi estimado que por volta de 90% do tráfego total de *e-mails* eram *spams*.

TÉCNICAS DE SPAM

O SPAMMER FALSIFICA O REMETENTE DO EMAIL, OU SEJA, MASCARA O VERDADEIRO REMETENTE

SPOOFING

O SPAMMER SE PASSA POR EMPRESAS REAIS PARA CONSEGUIR SUAS INFORMAÇÕES DE LOGIN.

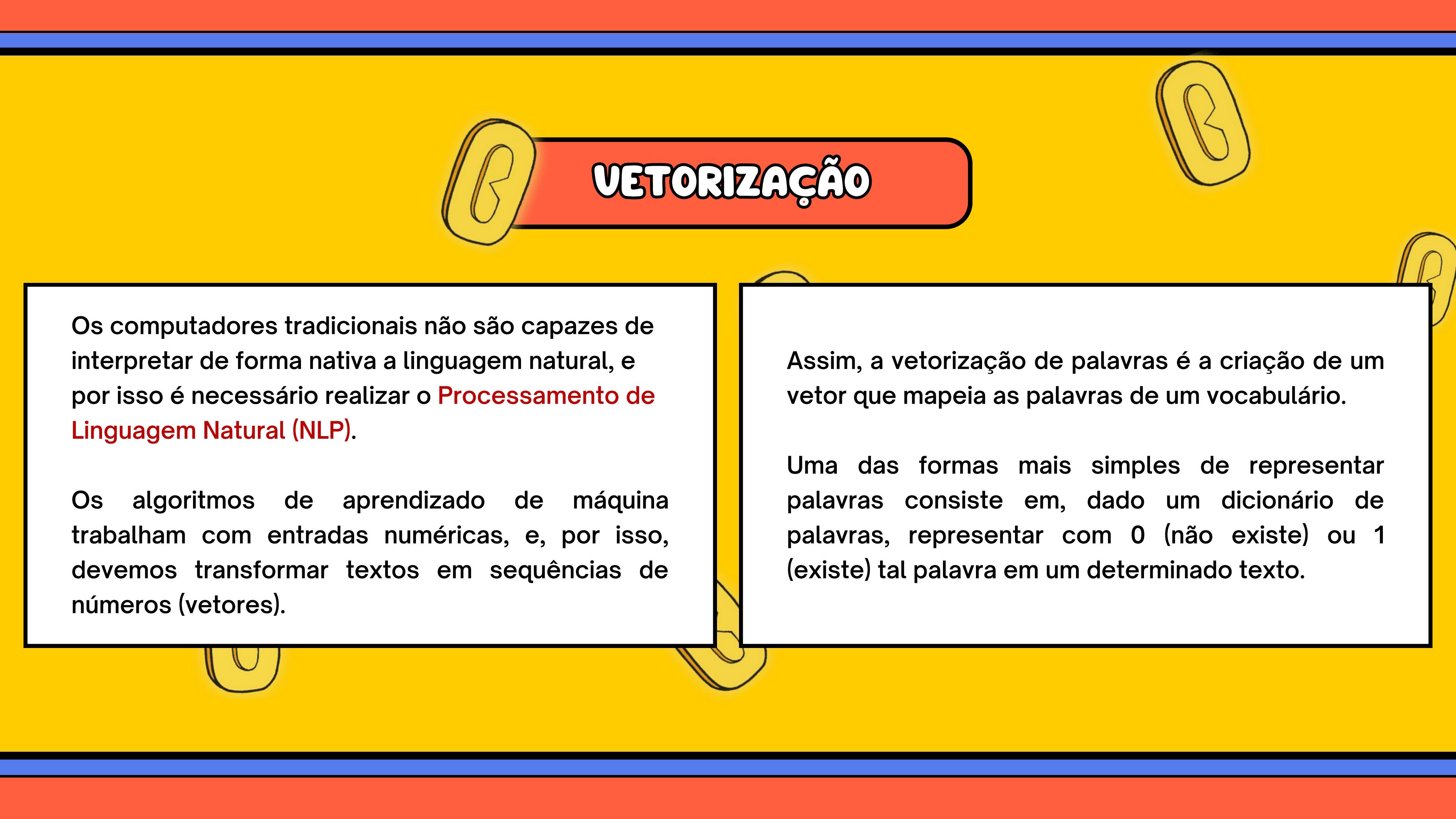
PHISHING

O SPAMMER INCORPORA MENSAGENS TEXTUAIS EM IMAGENS, PASSANDO PELOS DETECTORES DE SPAM TEXTUAIS.

IMAGE SPAM

É UM TIPO DE SPEAR PHISHING, PORÉM VOLTADO A "GRANDES ALVOS", COMO PESSOAS RELEVANTES OU DE ALTO CARGO.

WHALING



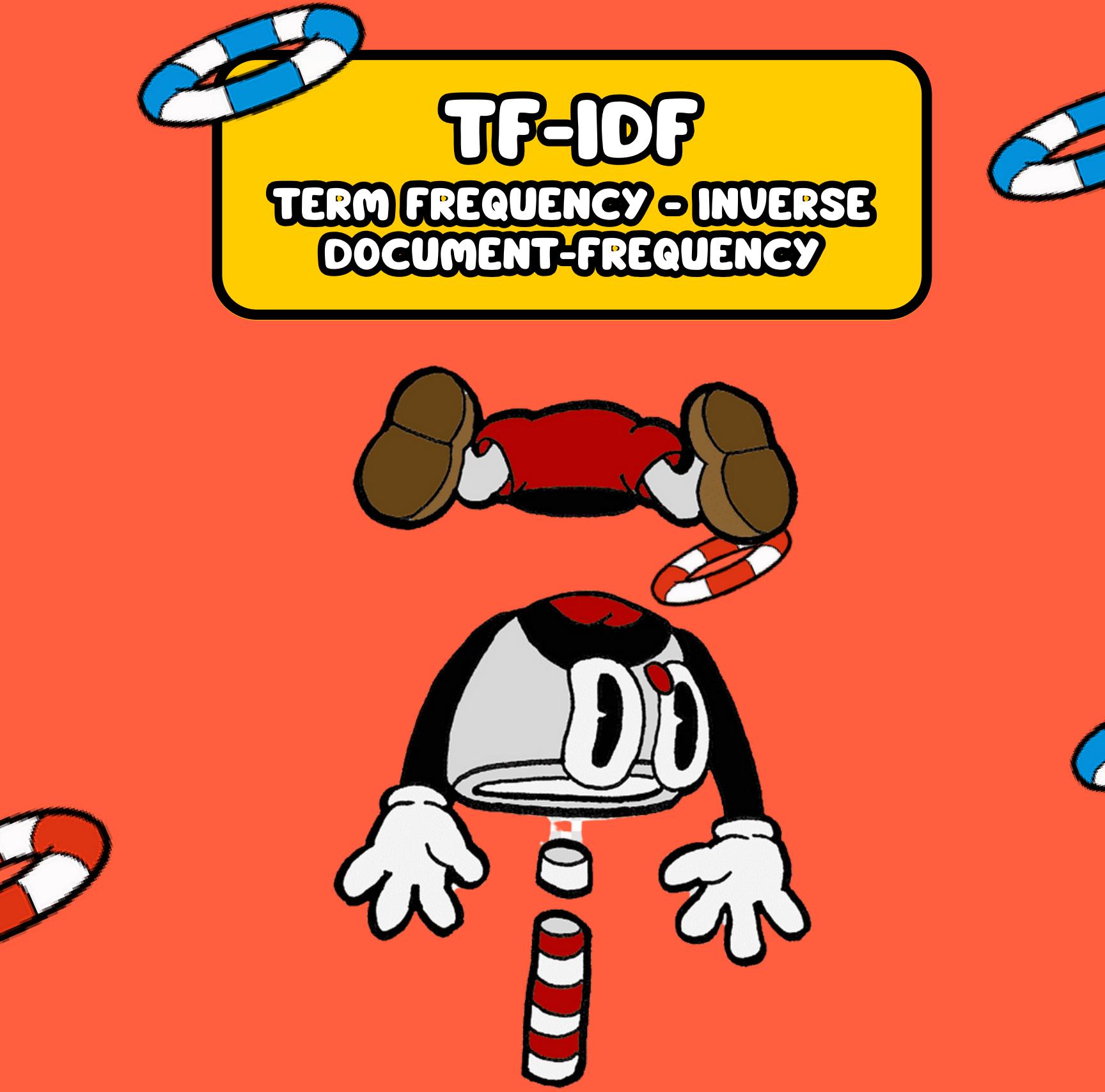
VETORIZAÇÃO

Os computadores tradicionais não são capazes de interpretar de forma nativa a linguagem natural, e por isso é necessário realizar o **Processamento de Linguagem Natural (NLP)**.

Os algoritmos de aprendizado de máquina trabalham com entradas numéricas, e, por isso, devemos transformar textos em sequências de números (vetores).

Assim, a vetorização de palavras é a criação de um vetor que mapeia as palavras de um vocabulário.

Uma das formas mais simples de representar palavras consiste em, dado um dicionário de palavras, representar com 0 (não existe) ou 1 (existe) tal palavra em um determinado texto.



TF-IDF

TERM FREQUENCY - INVERSE DOCUMENT-FREQUENCY

Uma outra técnica é a chamada *Term Frequency*, que é calculada levando em consideração a frequência de cada termo em um documento.

$$TF(p) = \frac{t_p}{T}$$

Uma variação desta técnica, chamada *Inverse Document-Frequency* atribui um peso maior às palavras mais raras entre os textos e, consequentemente, menores pesos às mais comuns:

$$IDF(p) = \log\left(\frac{D}{d_p}\right)$$

A técnica TF-IDF é obtida a partir do produto entre as duas equações anteriores, unindo as duas técnicas. Assim, quanto mais uma palavra ocorre em um documento, e quanto menos ela aparece entre todos os documentos, maior será seu valor.

TF-IDF

**TERM FREQUENCY - INVERSE
DOCUMENT-FREQUENCY**



EXEMPLO

Doc. 1: "It's going to rain today."

Doc. 2: "Today I am not going outside."

Doc. 3: "I am going to watch the season premiere."

Encontrando TF:

TF-IDF

TERM FREQUENCY - INVERSE DOCUMENT-FREQUENCY



Palavras	Doc. 1	Doc. 2	Doc. 3
going	0.16	0.16	0.12
to	0.16	0	0.12
today	0.16	0.16	0
i	0	0.16	0.12
am	0	0.16	0.12
it	0.16	0	0
is	0.16	0	0
rain	0.16	0	0

TF-IDF

**TERM FREQUENCY - INVERSE
DOCUMENT-FREQUENCY**



EXEMPLO

Doc. 1: "It's going to rain today."

Doc. 2: "Today I am not going outside."

Doc. 3: "I am going to watch the season premiere."

Encontrando TF:

Encontrando IDF:

TF-IDF

TERM FREQUENCY - INVERSE DOCUMENT-FREQUENCY



Palavras	IDF
going	$\log(3/3)$
to	$\log(3/2)$
today	$\log(3/2)$
i	$\log(3/2)$
am	$\log(3/2)$
it	$\log(3/1)$
is	$\log(3/1)$
rain	$\log(3/1)$

TF-IDF

**TERM FREQUENCY - INVERSE
DOCUMENT-FREQUENCY**



EXEMPLO

Doc. 1: "It's going to rain today."

Doc. 2: "Today I am not going outside."

Doc. 3: "I am going to watch the season premiere."

Encontrando TF:

Encontrando IDF:

Construindo o TF-IDF:

TF-IDF

TERM FREQUENCY - INVERSE DOCUMENT-FREQUENCY



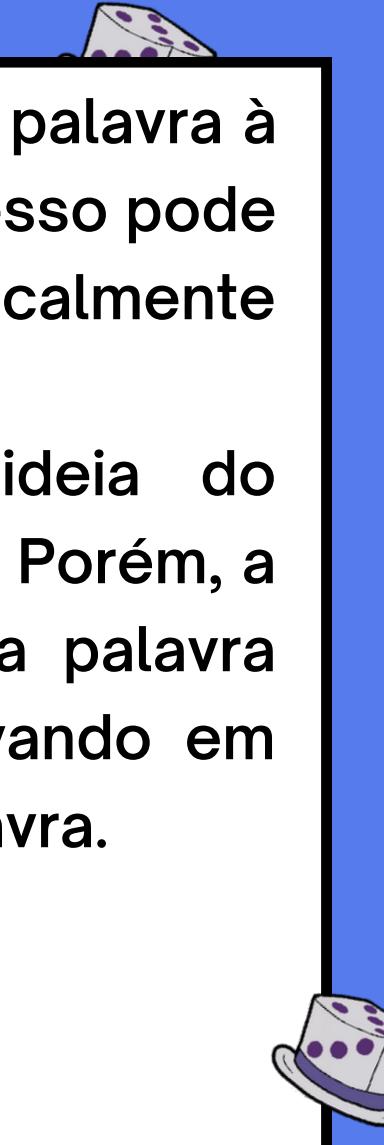
Palavras	Doc. 1	Doc. 2	Doc. 3
going	0	0	0
to	0.07	0	0.05
today	0.07	0.07	0
i	0	0.07	0.05
am	0	0.07	0.05
it	0.17	0	0
is	0.17	0	0
rain	0.17	0	0

PRÉ-PROCESSAMENTO



Quando falamos de NLP, geralmente os dados de entrada não estão estruturados da mesma forma, pois muitas vezes eles são adquiridos de diversas fontes. Assim, para tornar o processamento de dados mais eficiente, é interessante realizar o pré-processamento dos textos. Algumas técnicas de pré-processamento são:

- Remoção de stop words: stop words são as palavras mais comuns de uma linguagem e que podem ser suprimidas sem grandes perdas ao valor das frases;

- 
- *Stemming*: é o processo de reduzir cada palavra à sua raíz, ou seja, ao seu **stem**. Este processo pode reduzir uma palavra a outra gramaticalmente incorreta;
 - *Lemmatization*: consiste na mesma ideia do *stemming*, reduzir uma palavra à sua raiz. Porém, a redução aqui sempre resultará em uma palavra que realmente existe na gramática, levando em consideração a classe gramatical da palavra.

PRÉ-PROCESSAMENTO

Quando falamos de NLP, geralmente os dados

de entrada não estão
pois muitas vezes e
fontes. Assim, para t
mais eficiente, é
processamento dos
processamento são:

- Remoção de stop words;
palavras mais comuns
podem ser suprimidas
valor das frases;

Original	Stemming	Lematization
was	wa	be
studies	studi	study
studying	study	study

Stemming é o processo de reduzir cada palavra à sua raiz.

Este processo pode ser gramaticalmente incorreto.

A lematização é a mesma ideia do stemming, mas aponta para a raiz gramaticalmente correta em uma palavra.

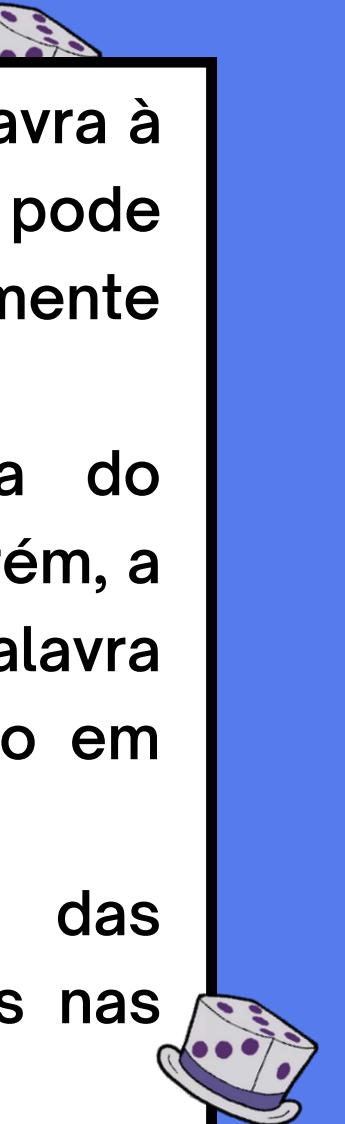


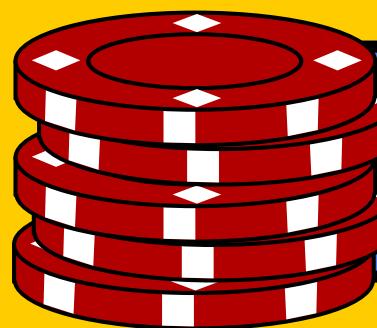
PRÉ-PROCESSAMENTO



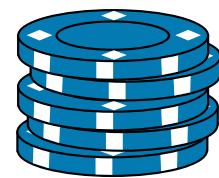
Quando falamos de NLP, geralmente os dados de entrada não estão estruturados da mesma forma, pois muitas vezes eles são adquiridos de diversas fontes. Assim, para tornar o processamento de dados mais eficiente, é interessante realizar o pré-processamento dos textos. Algumas técnicas de pré-processamento são:

- Remoção de *stop words*: *stop words* são as palavras mais comuns de uma linguagem e que podem ser suprimidas sem grandes perdas ao valor das frases;

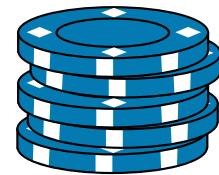
- 
- *Stemming*: é o processo de reduzir cada palavra à sua raíz, ou seja, ao seu ***stem***. Este processo pode reduzir uma palavra a outra gramaticalmente incorreta;
 - *Lemmatization*: consiste na mesma ideia do *stemming*, reduzir uma palavra à sua raiz. Porém, a redução aqui sempre resultará em uma palavra que realmente existe na gramática, levando em consideração a classe gramatical da palavra;
 - Remoção de pontuação: a remoção das pontuações permite nos concentrar apenas nas palavras de um documento.



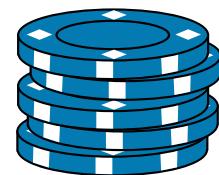
SVM - SUPPORT VECTOR MACHINE



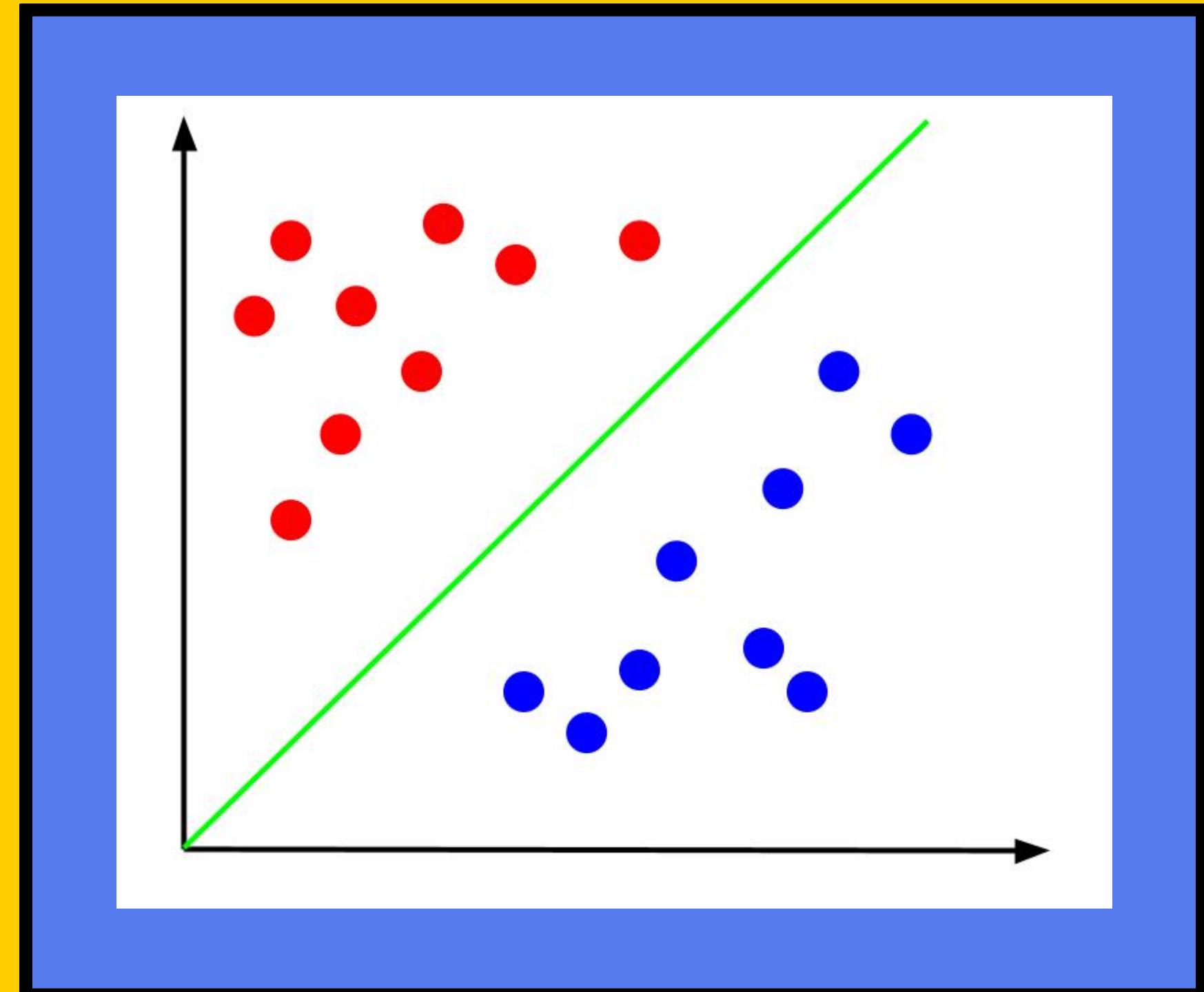
Máquina de vetores de suporte (SVM) é um algoritmo de aprendizado de máquina supervisionado utilizada para classificação.



O SVM determina um limite de decisão dividindo o espaço em dois subespaços: um espaço para cada classe (*spam* ou *ham*).



No caso de NLP, temos que cada uma das palavras de um documento é uma dimensão, e cada valor obtido pelo TF-IDF é o valor naquela dimensão.



O classificador Naive Bayes é uma solução de aprendizado de máquina que classifica os dados a partir de uma tabela de probabilidades. Funciona baseado no princípio do teorema de Bayes de que todas as características são independentes e igualmente importantes.

Segundo o teorema de Bayes e adaptando para o NLP, temos que, dado um vetor de palavras \mathbf{W} , a probabilidade de pertencer a uma classe \mathbf{C} é dada por:

$$p(C_k | \mathbf{W}) = \frac{p(C_k) \times p(\mathbf{W} | C_k)}{p(\mathbf{W})}$$



E que podemos escrever como:

$$\text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}$$

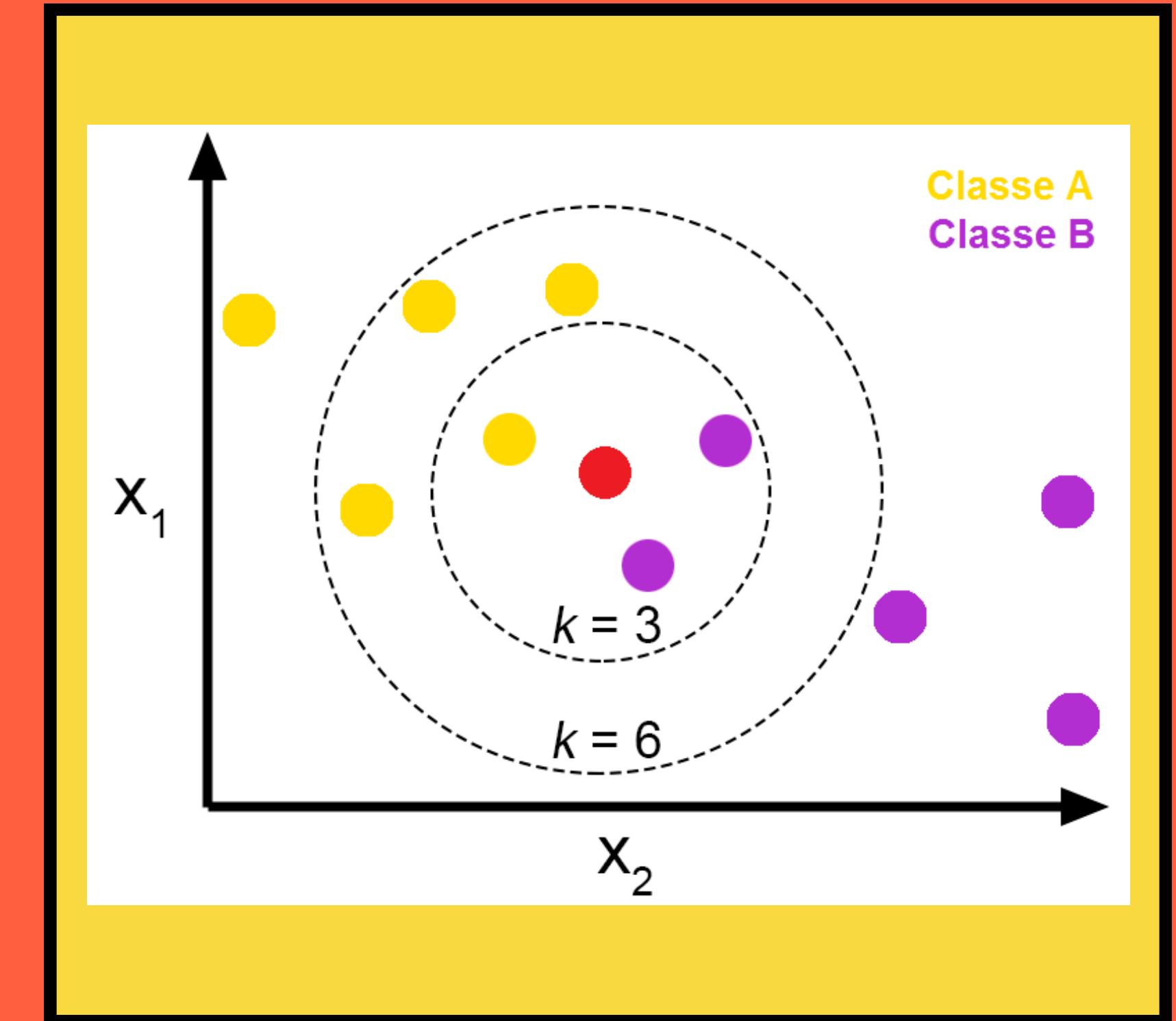
Neste trabalho, utilizamos o algoritmo de Naive Bayes Multinomial. O algoritmo Multinomial é usado quando estamos lidando com valores discretos, que, no nosso caso, são os pesos da aplicação do TF-IDF nos documentos. Assim, temos que a probabilidade $p(\mathbf{W})$ será dada pelos valores obtidos através da aplicação do TF-IDF.

KNN

O kNN é um algoritmo de aprendizado de máquina supervisionado utilizado para classificar componentes e seus vizinhos.

A classificação utilizando este modelo é realizada levando em consideração as classes de cada um dos k vizinhos mais próximos, como se fosse um sistema de votação.

Se mais vizinhos são classificados em uma determinada classe, então o texto em questão será classificado pertencendo a esta classe também.



BERT

BIDIRECTIONAL ENCODER REPRESENTATIONS FOR TRANSFORMERS



BERT é uma técnica de pré-treinamento para NLP. É o resultado de pesquisas da Google sobre os **Transformers**, que são modelos que processam palavras relacionando-as com todas as outras em uma sentença, ao invés de relacioná-las uma por uma.

O Transformer é uma arquitetura de rede neural baseada em um mecanismo de autoatenção. Transformers foram desenvolvidos para resolver o problema de tradução por rede neural. Isso significa que, para toda entrada, o modelo irá produzir uma saída.

Por exemplo, consideremos as seguintes sentenças:

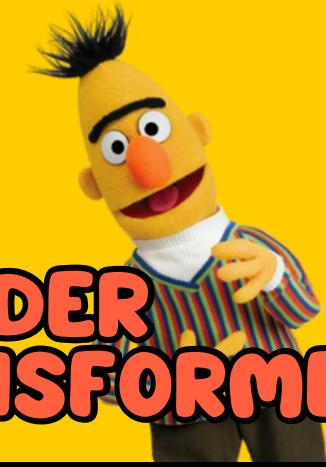
- *bank account*;
- *bank of the river*.

Claramente podemos observar que a palavra *bank* não possui o mesmo significado nas duas sentenças. Portanto, em um processo de tradução, o modelo deve entender isso, levando em consideração dependências e conexões entre as palavras.

Para isso, Redes Neurais Recorrentes e Convolucionais estavam sendo utilizadas, mas possuíam alguns problemas.

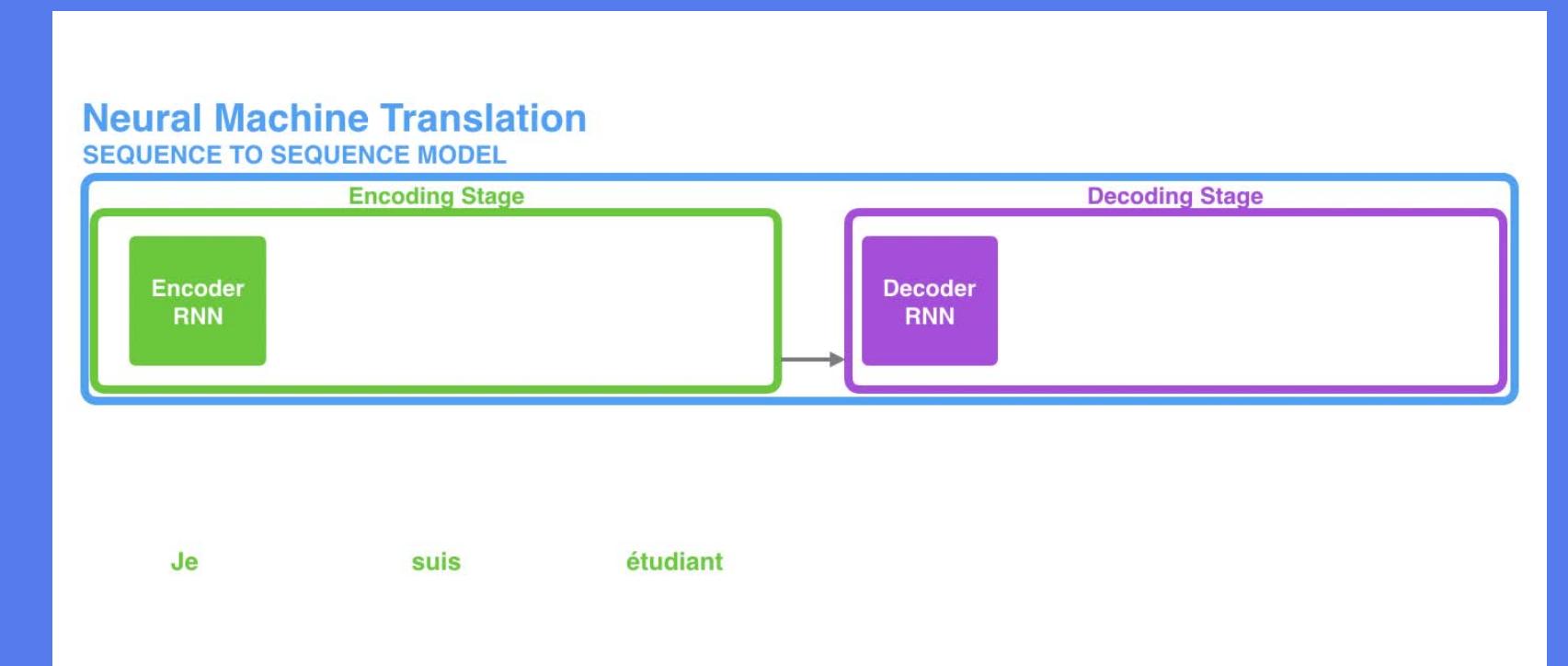
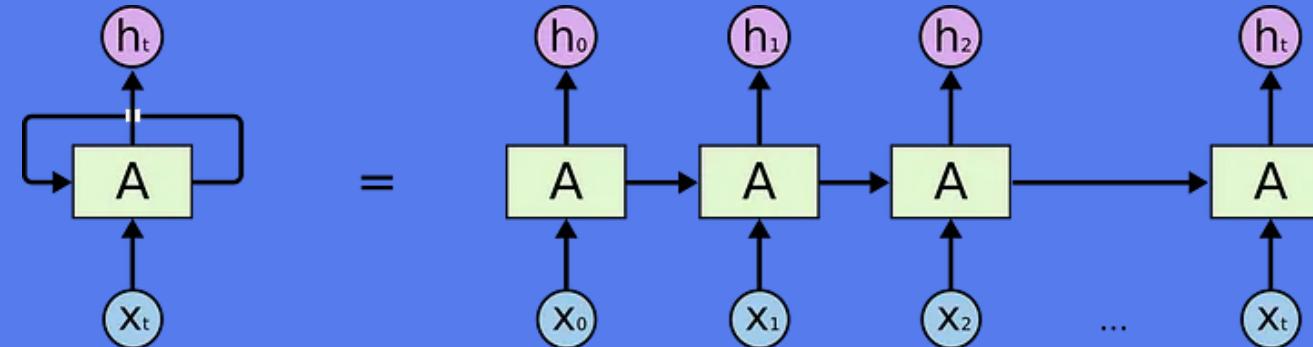
BERT

BIDIRECTIONAL ENCODER REPRESENTATIONS FOR TRANSFORMERS



REDES NEURAIS RECORRENTES

Redes Neurais Recorrentes possuem loops em si, permitindo que as informações persistam. Podemos pensar também de uma forma diferente. Uma RNN pode ser pensada como múltiplas cópias da mesma rede, em que cada rede passa uma mensagem para seu sucessor.



BERT

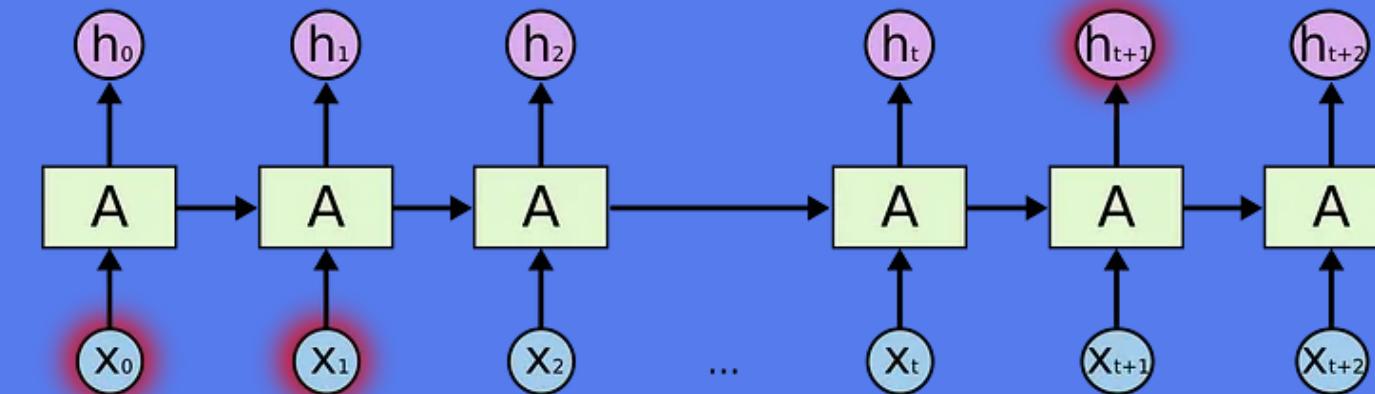
BIDIRECTIONAL ENCODER
REPRESENTATIONS FOR TRANSFORMERS



REDES NEURAIS RECORRENTES

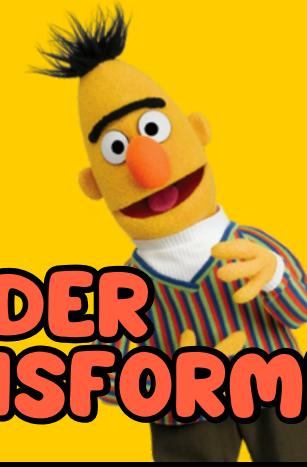
O problema das RRNs são as dependências de longo prazo. Quando estamos tratando textos pequenos, as RRNs conseguem facilmente transmitir a informações entre as redes.

Mas existem casos em que precisa-se de mais contexto. RRNs se tornam ineficazes quando existe um espaço muito grande entre informações relevantes. Assim, quanto maior a distância entre as palavras, maior a chance de que a informação seja perdida.



BERT

BIDIRECTIONAL ENCODER REPRESENTATIONS FOR TRANSFORMERS

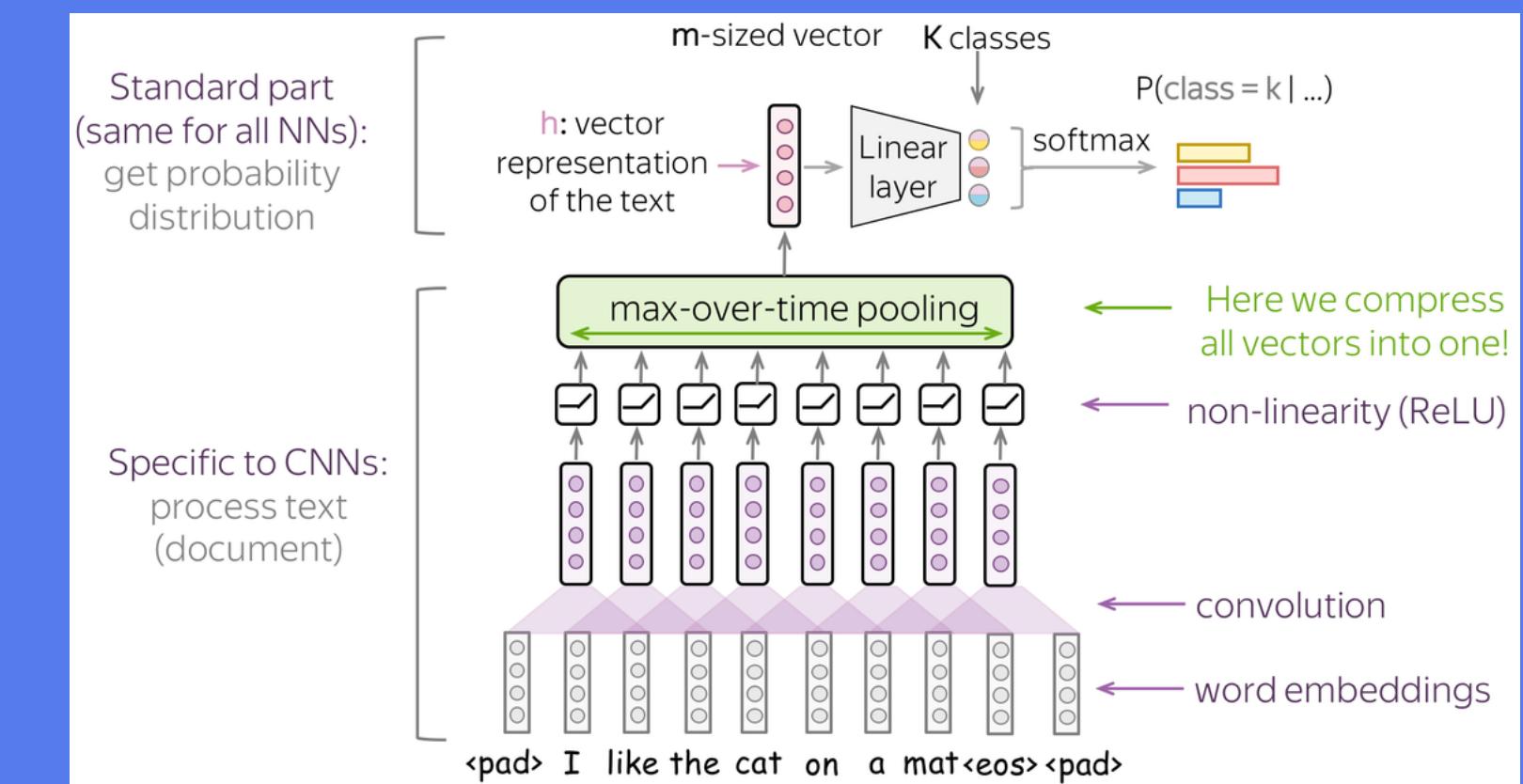


REDES NEURAIS CONVOLUCIONAIS

As Redes Neurais Convolucionais resolvem esse problema. As CNNs permitem:

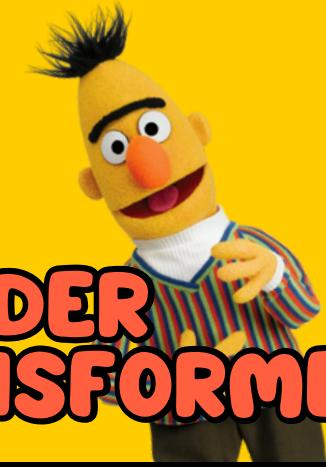
- paralelizar as entradas;
- explorar dependências locais;
- distância entre as palavras é logarítmica.

Porém, como veremos a seguir, a ordem das palavras não faz tanta diferença para o algoritmo, o que sabemos que, no mundo real, isso é uma característica importante. Por isso, as CNNs ainda possuem limitações.



BERT

BIDIRECTIONAL ENCODER
REPRESENTATIONS FOR TRANSFORMERS

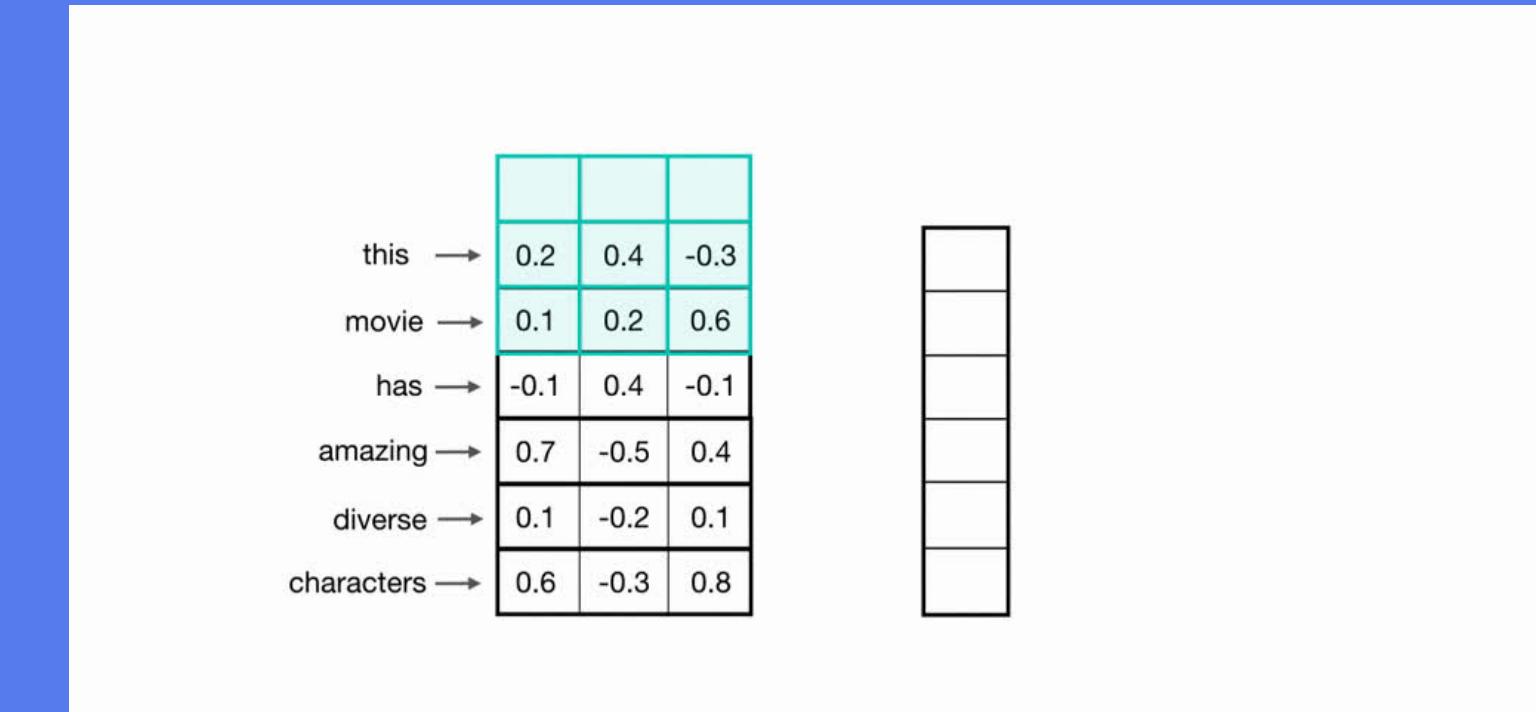


REDES NEURAIS CONVOLUCIONAIS

Embedding: é o processo de vetorização de palavras. A aplicação dessa técnica identifica relação entre as palavras a partir de sua vizinhança, contexto específico das palavras. Um dos métodos mais comuns é o **Word2Vec**.

Convolução: é um processo de redução de dimensionalidade para encontrar padrões. Assim, dado um kernel ou filtro, os valores obtidos pelo Embedding são atualizados.

this →	0.2	0.4	-0.3
movie →	0.1	0.2	0.6
has →	-0.1	0.4	-0.1
amazing →	0.7	-0.5	0.4
diverse →	0.1	-0.2	0.1
characters →	0.6	-0.3	0.8



BERT

BIDIRECTIONAL ENCODER
REPRESENTATIONS FOR TRANSFORMERS



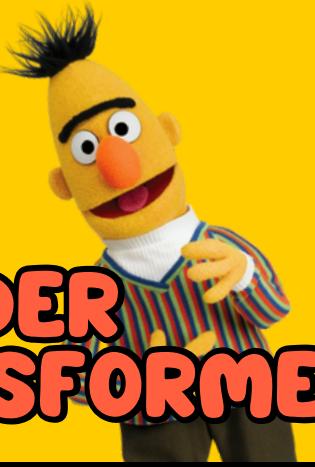
REDES NEURAIS CONVOLUCIONAIS

Esse processo que vimos acima é realizado para cada kernel da CNN. Quanto maior o kernel, mais palavras da frase são analisadas ao mesmo tempo, gerando diferentes resultados. Independente do tamanho do kernel, o output sera um único valor obtido pelo pooling. Esses valores são concatenados para se transformar novamente em um vetor único. Esse vetor final é utilizado como camada de entrada de neurônios da rede neural que será treinada.

A vantagem da CNN é que não é necessário utilizar todo um texto como camada de entrada, apenas as características que foram extraídas pelos filtros, fazendo com que o processo seja mais rápido e interativo.

BERT

BIDIRECTIONAL ENCODER
REPRESENTATIONS FOR TRANSFORMERS



TRANSFORMERS

Os Transformers podem ser divididos em Codificadores e Decodificadores. Os codificadores possuem duas subcamadas: camada de autoatenção e a Rede Neural (Feed Forward NN). O que é essa camada de autoatenção?

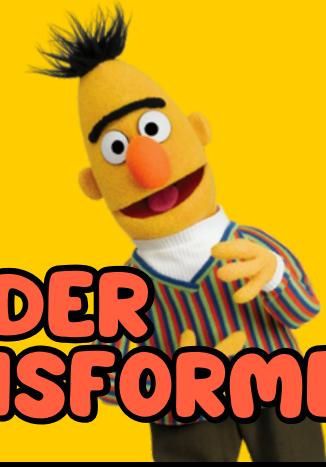
Digamos que queremos traduzir a seguinte frase: "The animal didn't cross the street because it was too tired". A palavra "it" se refere à "street" ou à "animal"?

À medida que o modelo processa cada palavra, a autoatenção permite que ele examine outras posições na sequência de entrada em busca de pistas que possam ajudar a melhorar a codificação dessa palavra.

A autoatenção é o método que o Transformer usa para incorporar a "compreensão" de outras palavras relevantes naquela que estamos processando no momento.

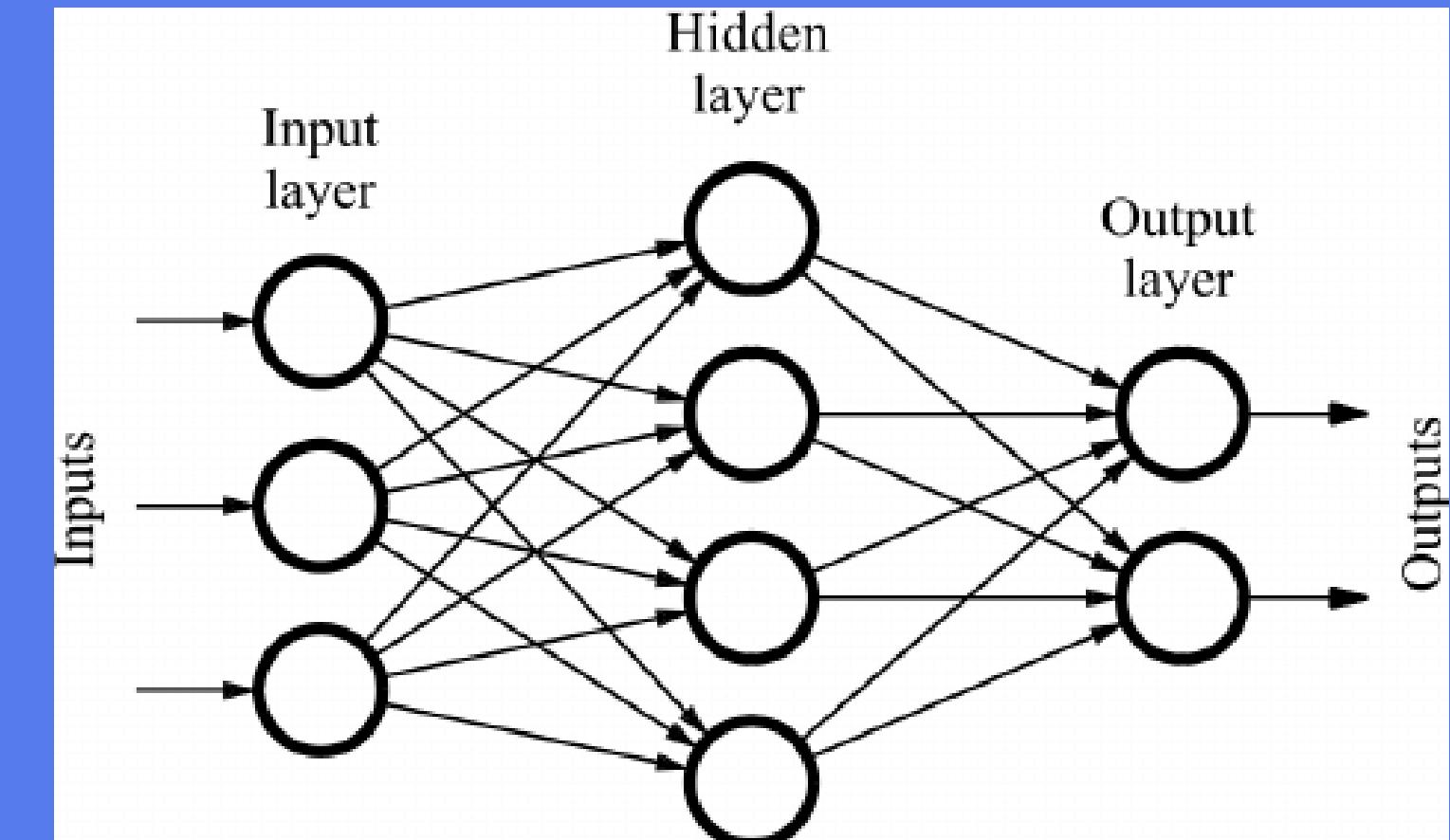
BERT

BIDIRECTIONAL ENCODER REPRESENTATIONS FOR TRANSFORMERS



REDES NEURAIS FEED FORWARD

Uma Rede Neural Feed Forward é uma rede neural artificial em que as conexões entre os nós não formam um ciclo, ou seja, a informação é processada sempre em um único sentido. É o oposto do que chamamos de Rede Neural Recorrente.



BERT

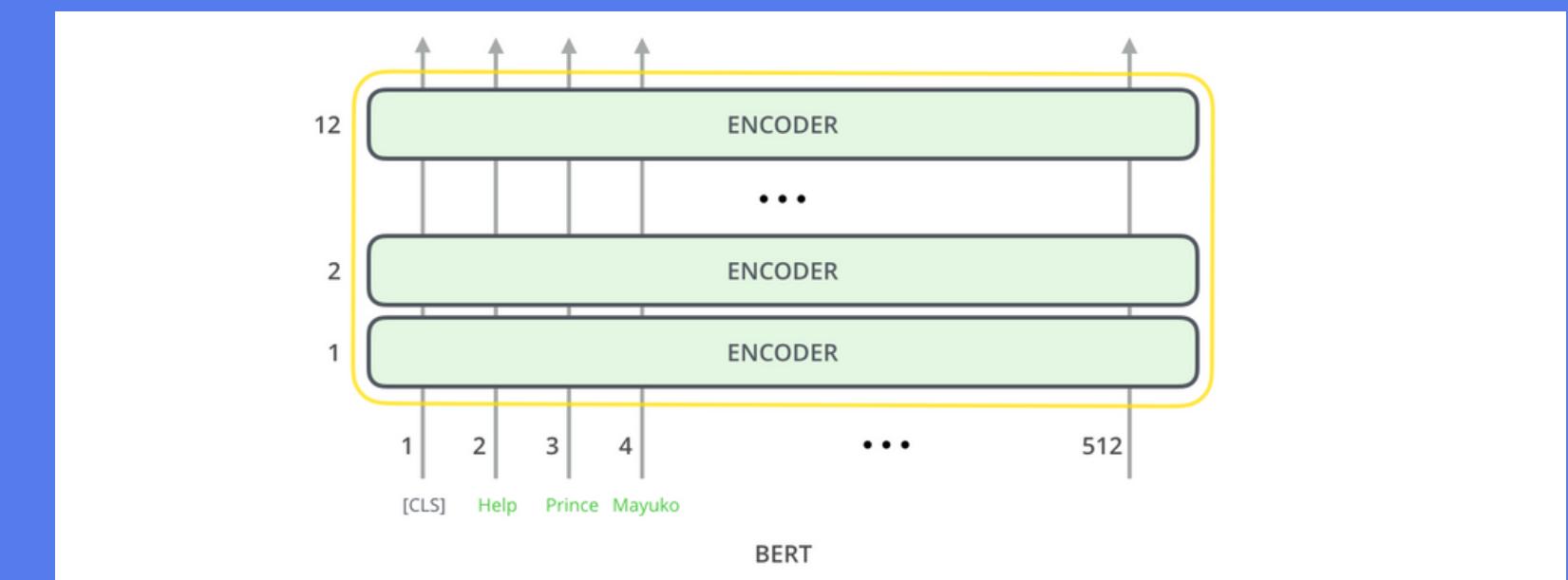
BIDIRECTIONAL ENCODER
REPRESENTATIONS FOR TRANSFORMERS



E FINALMENTE... BERT!

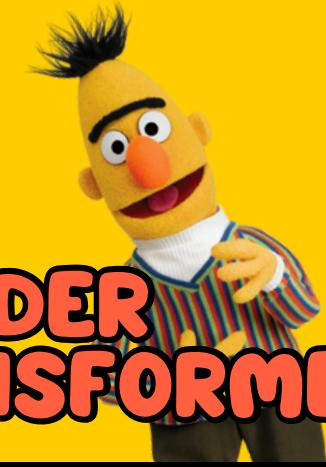
Proposto em 2018, o lançamento do BERT significou o início de uma nova era na NLP, quebrando diversos recordes de modelos que lidavam com tarefas baseadas em linguagens.

O BERT é basicamente uma pilha de codificadores de Transformers. Portanto, a entrada do BERT é idêntica às entradas nos Transformers, em que, recebendo uma sequência de palavras, cada codificador aplica autoatenção e passa seus resultados por meio de um rede neural feed forward. A diferença é nas saídas.



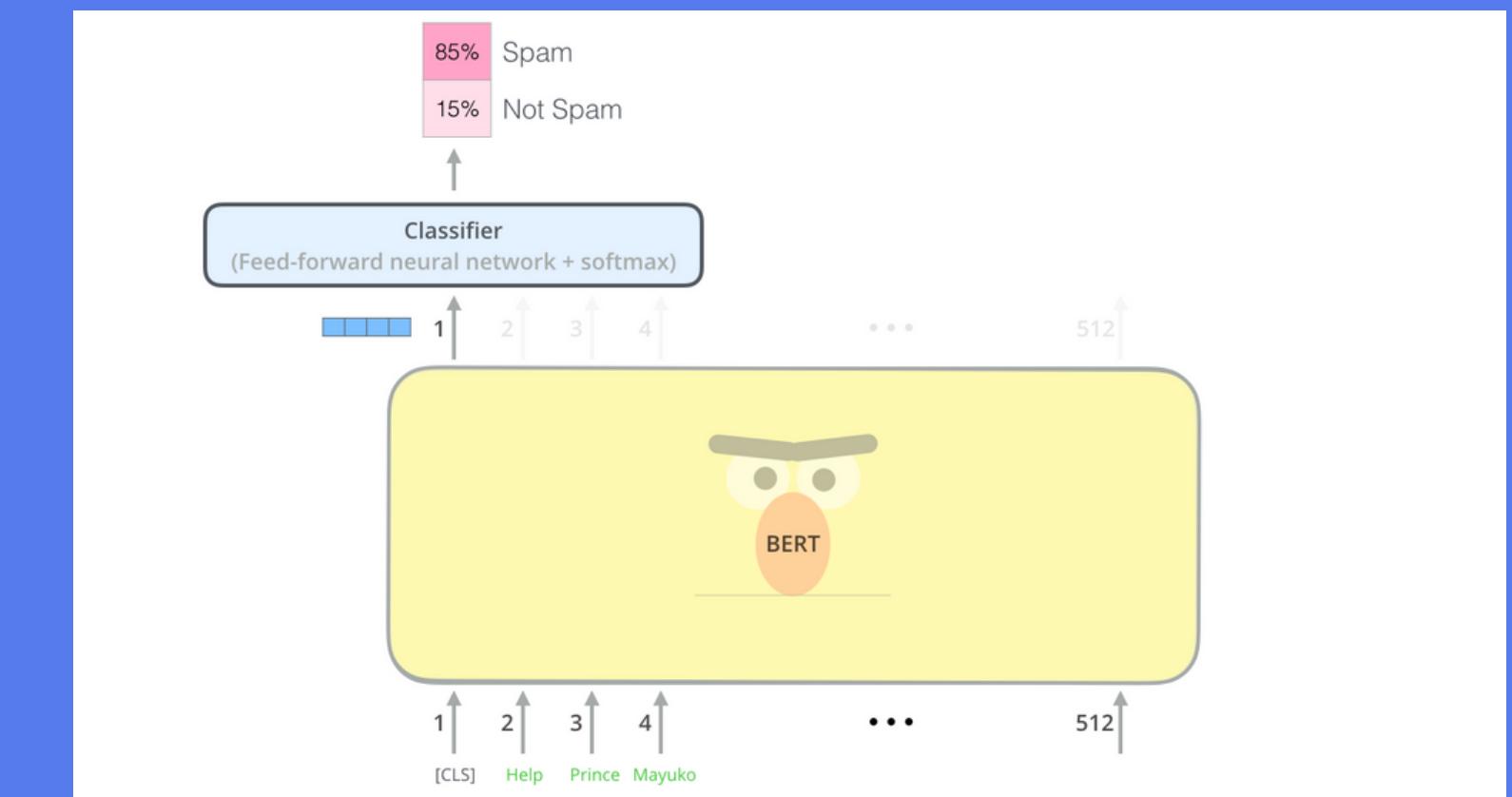
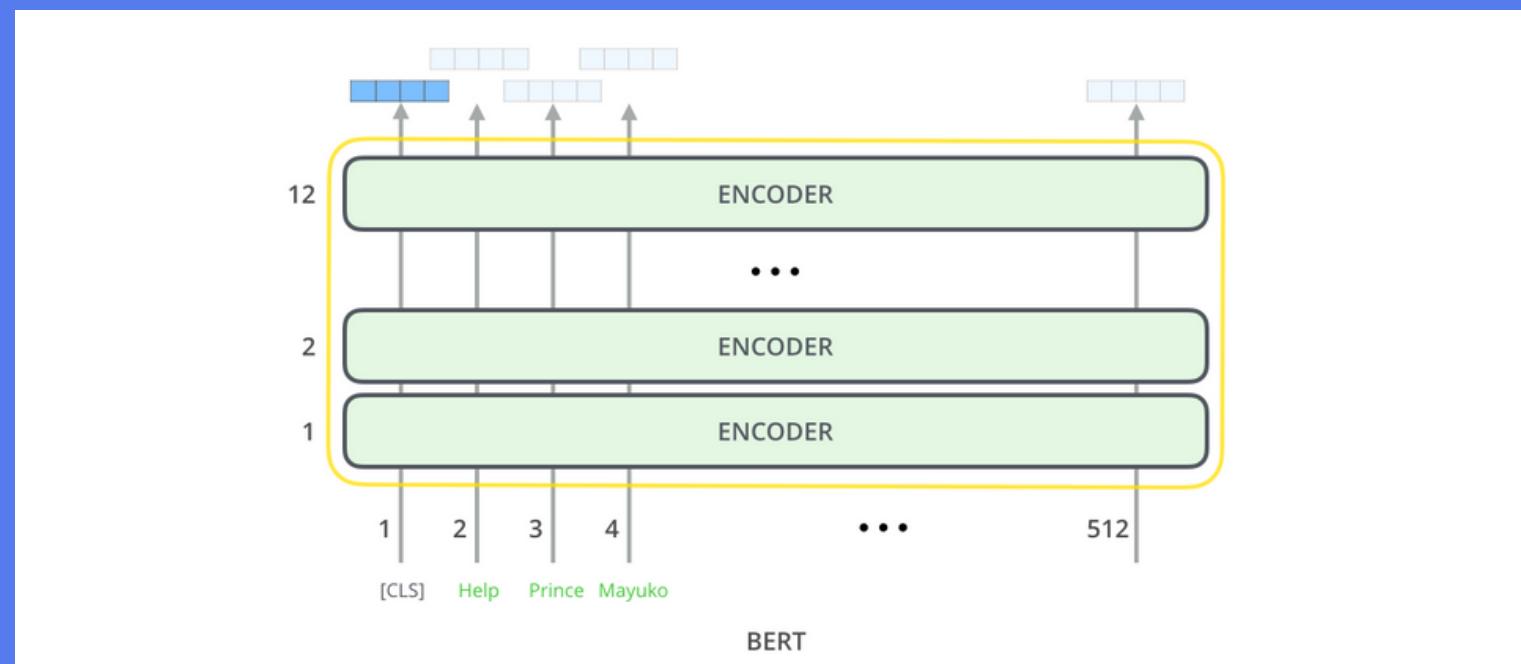
BERT

BIDIRECTIONAL ENCODER REPRESENTATIONS FOR TRANSFORMERS



E FINALMENTE... BERT!

Cada palavra produz um vetor que agora pode ser passado como entrada para um classificador de aprendizado de máquina de nossa escolha.



Assim, podemos dizer que o BERT é um modelo que processa palavras levando em consideração todas as outras naquelas sentenças.

BERT

**BIDIRECTIONAL ENCODER
REPRESENTATIONS FOR TRANSFORMERS**



DISTILBERT

O DistilBERT é uma versão menor, mais rápida, mais barata e mais leve do BERT desenvolvida pela HuggingFace. Essa versão alcança 97% do desempenho do BERT.

Essa versão é necessária porque técnicas como o BERT são técnicas custosas, e que necessitam de servidores caros para funcionarem. Assim, para resolver essa necessidade, surgiram técnicas como a destilação, que consiste em comprimir um modelo grande, o professor, em um modelo menor, o aluno.

REGRESSÃO LOGÍSTICA

É um modelo estatístico que utiliza a função logística a seguir:

$$f(x) = \frac{1}{1 + e^{-x}}$$

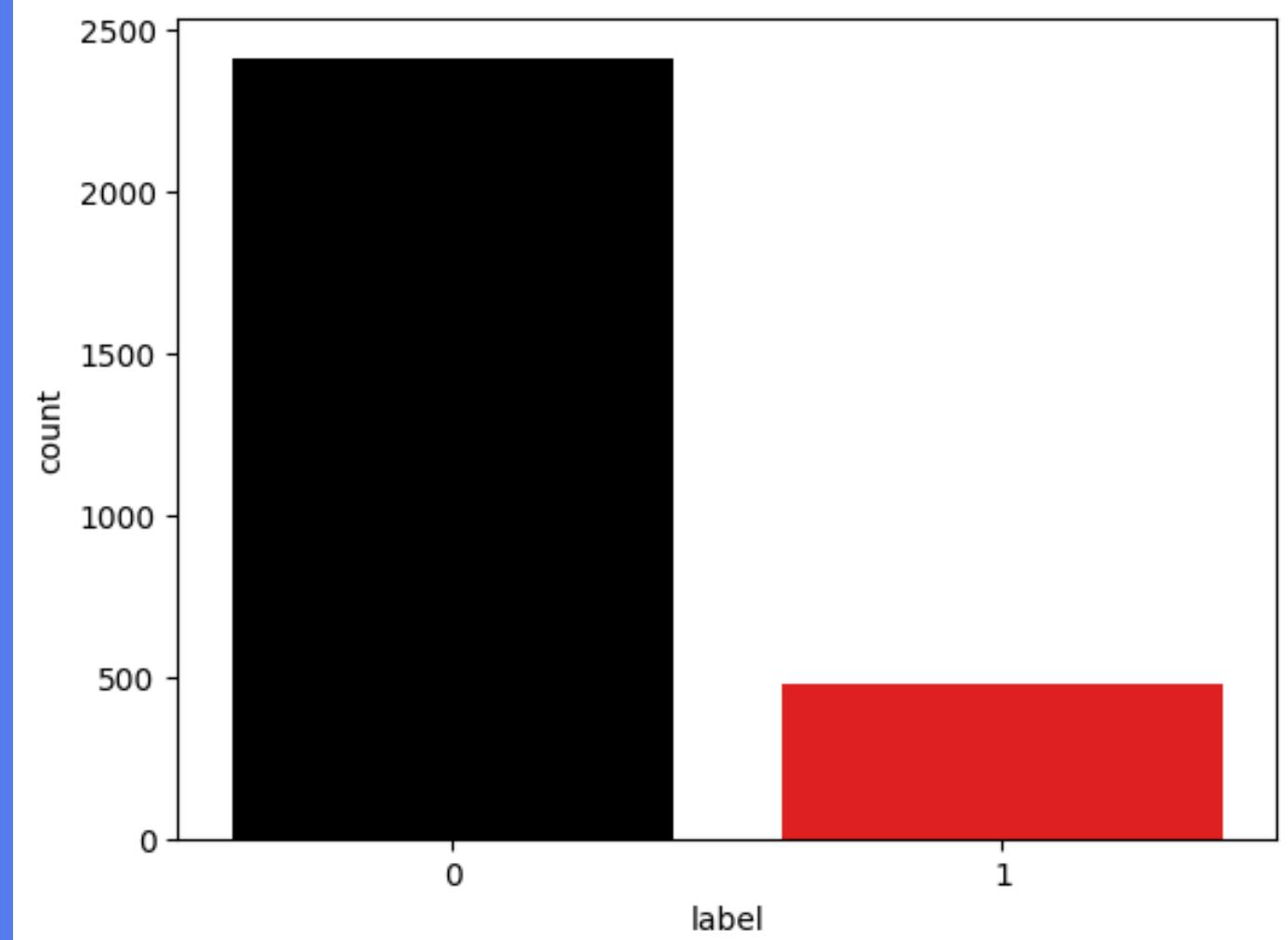
A função nos dá um valor entre zero e 1, que significa a probabilidade de pertencer a uma determinada classe. A diferença da Regressão Logística para a Regressão Linear é que, enquanto a Linear nos dá um valor contínuo, a Logística prevê a probabilidade de pertencer a uma classe ou não.

IMPLEMENTAÇÃO E RESULTADOS

Neste trabalho, iremos performar [Detecção de Spam em Emails](#).

Para esta tarefa, iremos utilizar o *dataset Lingspam*, que pode ser encontrado na plataforma Kaggle. A distribuição dos dados pode ser visualizada no gráfico ao lado. A classe 0 são os emails classificados como ham, e a classe 1 são os emails classificados como spam.

	subject	message	label
0	job posting - apple-iss research center	content - length : 3386 apple-iss research cen...	0
1		NaN lang classification grimes , joseph e . and ba...	0
2	query : letter frequencies for text identifica...	i am posting this inquiry for sergei atamas (...	0
3		risk a colleague and i are researching the differin...	0
4	request book information	earlier this morning i was on the phone with a...	0



PRÉ-PROCESSAMENTO E TOKENIZAÇÃO

No pré-processamento, realizamos as seguintes atividades:

- substituição dos valores nulos pela moda;
- truncamento do assunto com a mensagem;
- remoção da quebra de linha (\n);
- substituição de símbolos (\$ -> "money");
- substituição de números de telefone (303 059 0101 -> "phonenumbers");
- substituição de números (9 -> "number");
- remoção de caracteres especiais (exceto '@');
- remoção de stop-words;
- lematização.

Após isso, foi realizada a tokenização com o TF-IDF.

IMPLEMENTAÇÃO E RESULTADOS

sociolinguistic lang classification grime joseph e barbara f grime ethnologue language family index pb isbn phonenumbers number number vi number pp money number number summer institute linguistics companion volume ethnologue language world twelfth edition list language family world sub group show tree arrangement broad classification language family language family index facilitate locate language name ethnologue make datum accessible internet academic book @ sil org language reference lang culture gregerson marilyn ritual belief kinship sulawesi pb isbn phonenumbers number number ix number pp money number number summer institute linguistic seven article discuss language group sulawesi indonesia primary focus cultural matter linguistic content topic include traditional religion belief certain ceremony kinship internet academic book @ sil org language society indonesia computer le weber david j stephen r mcconnell diana weber beth j bryson primer tool develop early reading material pb isbn phonenumbers number number xvi number pp ms dos software money number number summer institute linguistics author present computer program instruction develop reading material language little background literacy book structure manual step step procedure establish appropriate primer sequence organize word phrase sentence correlate sequence presuppose thorough knowledge linguistic internet academic book @ sil org literacy computer

MODELOS E PREPARAÇÃO

Para avaliar os diferentes algoritmos, o *dataset* foi dividido em **70% treinamento** e **30% teste**. Como estamos lidando com um conjunto de dados desbalanceado, o *split* foi feito levando em consideração a proporção dos dados. Ou seja, em ambos conjunto de teste e treinamento, foi mantida essa proporção 70/30.

Para a execução dos algoritmos, iremos executá-los com diferentes hiperparâmetros e avaliar o desempenho no conjunto de treinamento pela medida **F1-Score**. Para a seleção dos melhores hiperparâmetros, será realizado um **3-fold cross validation**.

IMPLEMENTAÇÃO E RESULTADOS

Depois, com o melhor modelo, iremos calcular e avaliá-lo com as medidas:

- **F1-Score**;
- **Precision**;
- **Recall**;
- **Accuracy**.

SVM - SUPPORT VECTOR MACHINE

Realizando a execução do algoritmo SVM, utilizando o GridSearchCV(), testamos os seguintes parâmetros:

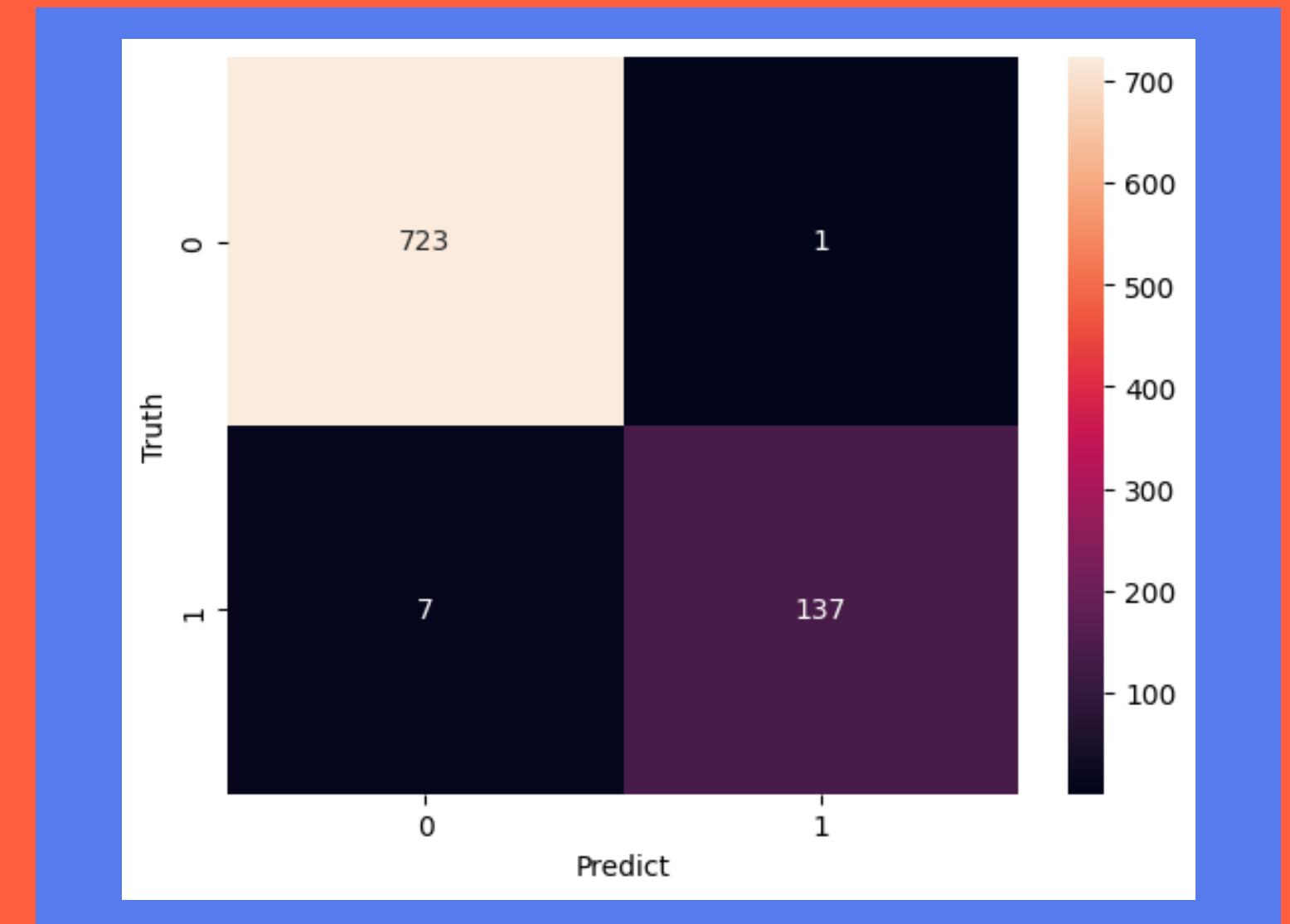
- kernel":['linear', 'poly', 'rbf', 'sigmoid'];
- C":[1, 3, 7].

A saída do GridSearchCV() nos dá que os melhores hiperparâmetros para o SVM são: C: 7, kernel: sigmoid. Testando o modelo com o conjunto de teste, temos a matriz de confusão que pode ser vista ao lado.

As medidas de avaliação foram:

- Accuracy: 0.9907834101382489
- Precision: 0.9911098840712547
- Recall: 0.9907834101382489
- F1 Score: 0.9908624426188452

IMPLEMENTAÇÃO E RESULTADOS



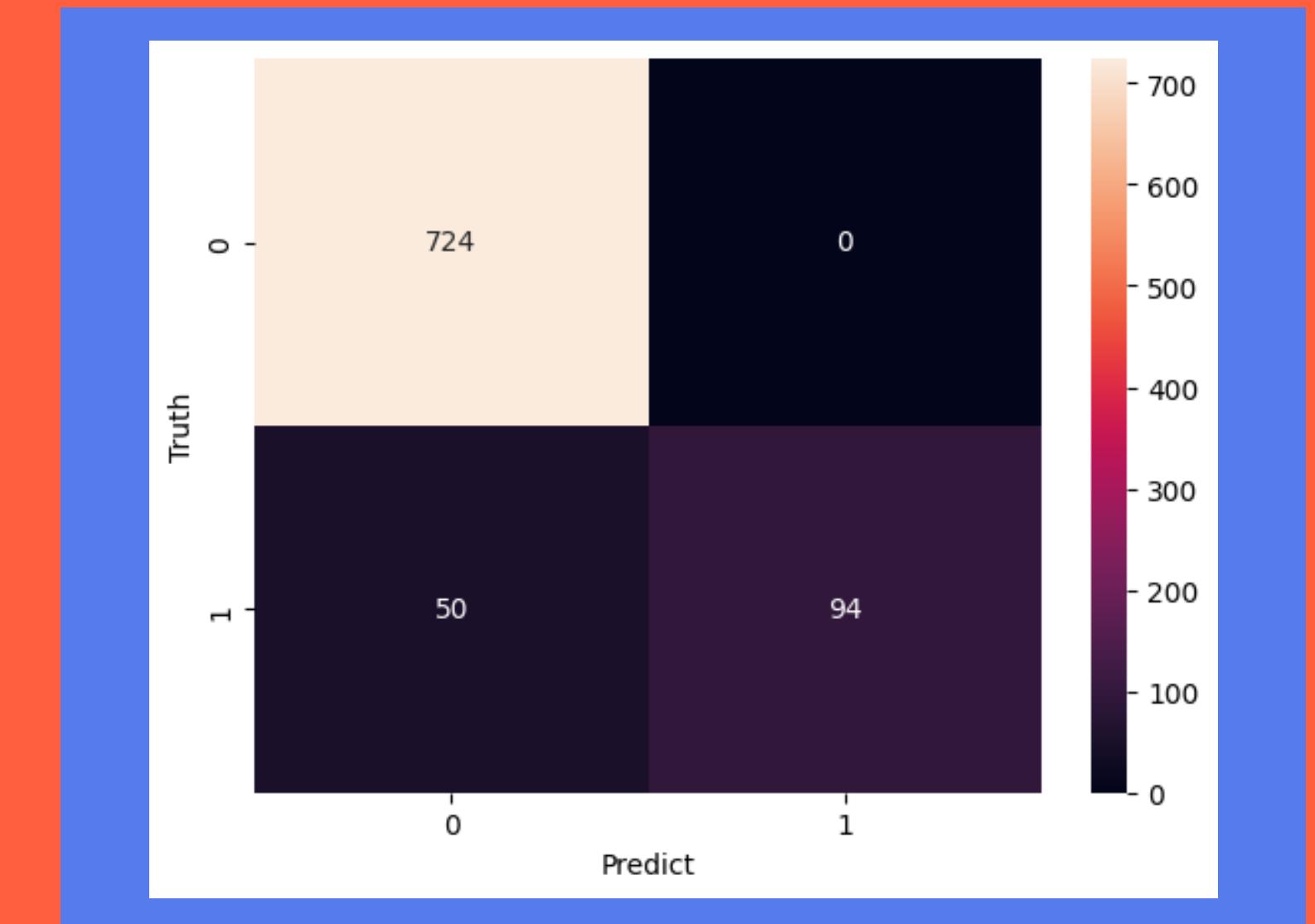
NAIVES BAYES

Realizando a execução do algoritmo Multinomial, temos as seguintes medidas de avaliação:

- Accuracy: 0.9423963133640553
- Precision: 0.9623975934459805
- Recall: 0.9423963133640553
- F1 Score: 0.9474857775488851

A matriz de confusão pode ser vista ao lado.

IMPLEMENTAÇÃO E RESULTADOS



KNN - K NEAREST NEIGHBORS

Realizando a execução do algoritmo kNN, utilizando o GridSearchCV(), testamos os seguintes parâmetros:

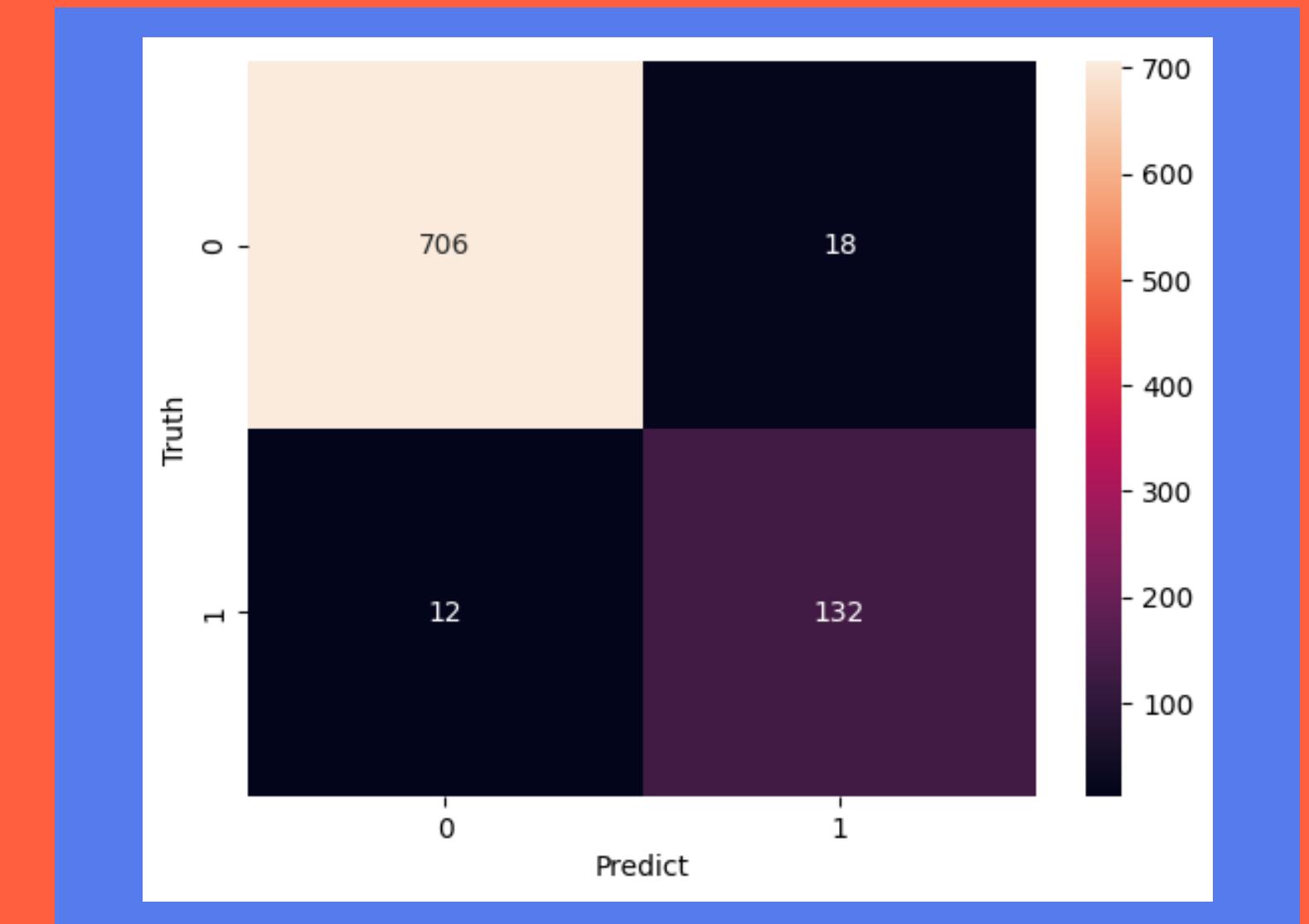
- n_neighbors: [1, 3, 5, 7];
- p: [1, 2].

A saída do GridSearchCV() nos dá que os melhores hiperparâmetros para o kNN são: n_neighbors: 7, p: 2. Testando o modelo com o conjunto de teste, temos a matriz de confusão que pode ser vista ao lado.

As medidas de avaliação foram:

- Accuracy: 0.9654377880184332
- Precision: 0.9650336074547445
- Recall: 0.9654377880184332
- F1 Score: 0.9651570171265678

IMPLEMENTAÇÃO E RESULTADOS



BERT

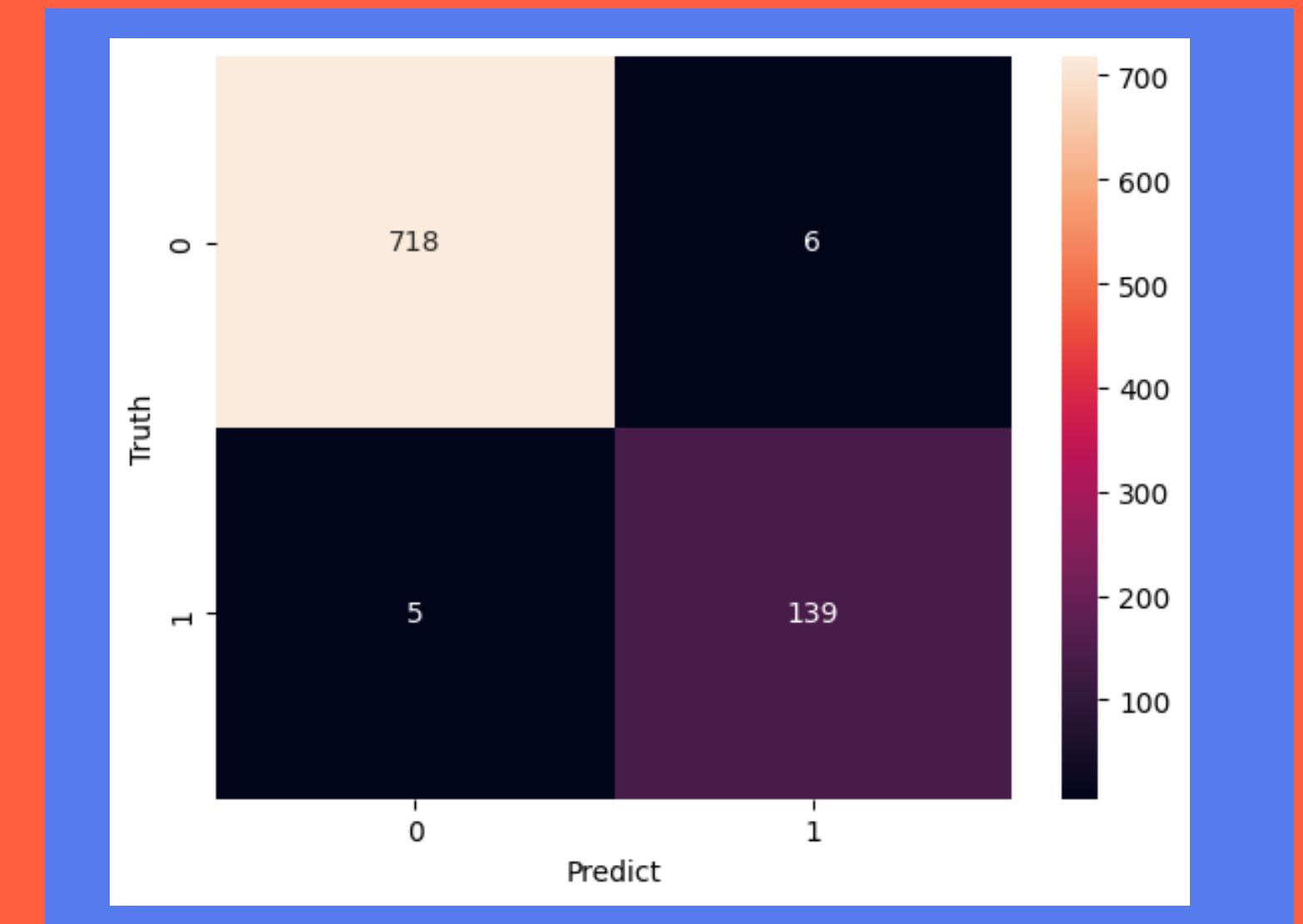
Após o pré-treinamento utilizando o BERT, utilizamos o modelo de Regressão Logística para classificar as instâncias. Realizando o GridSearchCV(), testamos o seguinte parâmetro: C: começando em 0.0001 até 100, gerar 20 amostras.

O melhor resultado foi C: 5.263253. A matriz de confusão pode ser observada ao lado.

As medidas de avaliação foram:

- Accuracy: 0.9873271889400922
- Precision: 0.9872967339522989
- Recall: 0.9873271889400922
- F1 Score: 0.9873096426562795

IMPLEMENTAÇÃO E RESULTADOS



FONTES BIBLIOGRÁFICAS

ALAMMAR, J. VISUALIZING A NEURAL MACHINE TRANSLATION MODEL (MECHANICS OF SEQ2SEQ MODELS WITH ATTENTION). DISPONÍVEL EM: <[HTTPS://JALAMMAR.GITHUB.IO/VISUALIZING-NEURAL-MACHINE-TRANSLATION-MECHANICS-OF-SEQ2SEQ-MODELS-WITH-ATTENTION/](https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/)>. ACESSO EM: 8 JUN. 2023A.

ALAMMAR, J. THE ILLUSTRATED TRANSFORMER. DISPONÍVEL EM: <[HTTP://JALAMMAR.GITHUB.IO/ILLUSTRATED-TRANSFORMER/](http://jalammar.github.io/illustrated-transformer/)>. ACESSO EM: 8 JUN. 2023B.

ALAMMAR, J. THE ILLUSTRATED BERT, ELMO, AND CO. (HOW NLP CRACKED TRANSFER LEARNING). DISPONÍVEL EM: <[HTTP://JALAMMAR.GITHUB.IO/ILLUSTRATED-BERT/](http://jalammar.github.io/illustrated-bert/)>. ACESSO EM: 8 JUN. 2023C.

ALAMMAR, J. A_VISUAL_NOTEBOOK_TO_USING_BERT_FOR_THE_FIRST_TIME.IPYNB AT MASTER · JALAMMAR/JALAMMAR.GITHUB.IO. [S.L: S.N.].

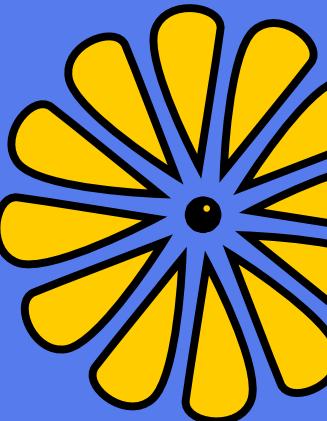
ALAMMAR, J. A VISUAL GUIDE TO USING BERT FOR THE FIRST TIME. DISPONÍVEL EM: <[HTTP://JALAMMAR.GITHUB.IO/A-VISUAL-GUIDE-TO-USING-BERT-FOR-THE-FIRST-TIME/](http://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/)>. ACESSO EM: 8 JUN. 2023E.

ANNA. SPAM_BERT.IPYNB AT MAIN · ANIASS/SPAM-DETECTION. [S.L: S.N.].

FONTES BIBLIOGRÁFICAS

ARIANE, G. O QUE É ATAQUE DE EMAIL SPOOFING E COMO RESOLVER? DISPONÍVEL EM: <[HTTP://WWW.HOSTINGER.COM.BR/TUTORIAIS/ATAQUE-DE-EMAIL-SPOOFING](http://WWW.HOSTINGER.COM.BR/TUTORIAIS/ATAQUE-DE-EMAIL-SPOOFING)>. ACESSO EM: 8 JUN. 2023.

EM:



BERT, THE SPAM DETECTOR THAT USES JUST 10 WORDS. DISPONÍVEL EM: <[HTTP://WWW.KAGGLE.COM/CODE/MRLUCASFISCHER/BERT-THE-SPAM-DETECTOR-THAT-US...](http://WWW.KAGGLE.COM/CODE/MRLUCASFISCHER/BERT-THE-SPAM-DETECTOR-THAT-US...)>. ACESSO EM: 8 JUN. 2023.

EM:

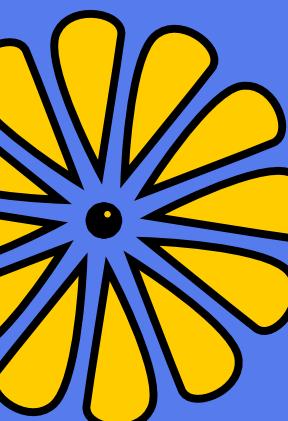
CARNEIRO, A. L. C. REDES NEURAIS CONVOLUCIONAIS PARA PROCESSAMENTO DE LINGUAGEM NATURAL. DISPONÍVEL EM: <[HTTPS://MEDIUM.COM/DATA-HACKERS/REDES-NEURAIS-CONVOLUCIONAIS-PARA-PROCESSAMENTO-DE-LINGUAGEM-NATURAL-935488D6901B](https://MEDIUM.COM/DATA-HACKERS/REDES-NEURAIS-CONVOLUCIONAIS-PARA-PROCESSAMENTO-DE-LINGUAGEM-NATURAL-935488D6901B)>. ACESSO EM: 8 JUN. 2023.

CONVOLUTIONAL MODELS FOR TEXT. DISPONÍVEL EM: <[HTTPS://LENA-VOITA.GITHUB.IO/NLP_COURSE/MODELS/CONVOLUTIONAL.HTML](https://LENA-VOITA.GITHUB.IO/NLP_COURSE/MODELS/CONVOLUTIONAL.HTML)>. ACESSO EM: 8 JUN. 2023.

EM: <[HTTPS://LENA-VOITA.GITHUB.IO/NLP_COURSE/MODELS/CONVOLUTIONAL.HTML](https://LENA-VOITA.GITHUB.IO/NLP_COURSE/MODELS/CONVOLUTIONAL.HTML)>

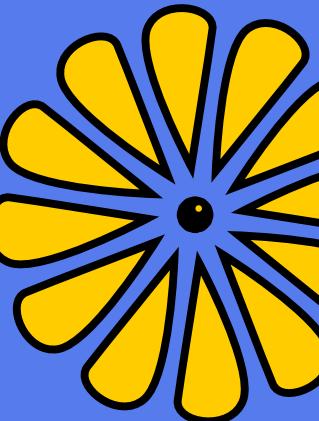
FEED FORWARD NEURAL NETWORK. DISPONÍVEL EM: <[HTTPS://DEEPAI.ORG/MACHINE-LEARNING-GLOSSARY-AND-TERMS/FEED-FORWARD-NEURAL-NETWORK](https://DEEPAI.ORG/MACHINE-LEARNING-GLOSSARY-AND-TERMS/FEED-FORWARD-NEURAL-NETWORK)>. ACESSO EM: 8 JUN. 2023.

FONSECA, C. WORD EMBEDDING: FAZENDO O COMPUTADOR ENTENDER O SIGNIFICADO DAS PALAVRAS. DISPONÍVEL EM: <[HTTPS://MEDIUM.COM/TURING-TALKS/WORD-EMBEDDING-FAZENDO-O-COMPUTADOR-ENTENDER-O-SIGNIFICADO-DAS-PALAVRAS-92FE22745057](https://MEDIUM.COM/TURING-TALKS/WORD-EMBEDDING-FAZENDO-O-COMPUTADOR-ENTENDER-O-SIGNIFICADO-DAS-PALAVRAS-92FE22745057)>. ACESSO EM: 8 JUN. 2023.



FONTES BIBLIOGRÁFICAS

GANESAN, K. WHAT ARE STOP WORDS? DISPONÍVEL EM: <[HTTP://KAVITA-GANESAN.COM/WHAT-ARE-STOP-WORDS/](http://KAVITA-GANESAN.COM/WHAT-ARE-STOP-WORDS/)>. ACESSO EM: 8 JUN. 2023.

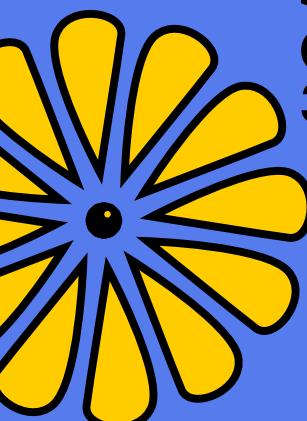


GIACAGLIA, G. HOW TRANSFORMERS WORK. DISPONÍVEL EM: <[HTTP://TOWARDSDATASCIENCE.COM/TRANSFORMERS-141E32E69591](http://TOWARDSDATASCIENCE.COM/TRANSFORMERS-141E32E69591)>. ACESSO EM: 8 JUN. 2023.

HOW CAN WE USE TFIDF VECTORS WITH MULTINOMIAL NAIVE BAYES? DISPONÍVEL EM: <[HTTPS://STACKOVERFLOW.COM/QUESTIONS/43237286/HOW-CAN-WE-USE-TFIDF-VECTORS-WITH-MULTINOMIAL-NAIVE-BAYES](https://STACKOVERFLOW.COM/QUESTIONS/43237286/HOW-CAN-WE-USE-TFIDF-VECTORS-WITH-MULTINOMIAL-NAIVE-BAYES)>. ACESSO EM: 8 JUN. 2023.

KARABIBER, F. TF-IDF — TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY – LEARNDATSCI. DISPONÍVEL EM: <[HTTPS://WWW.LEARNDATSCI.COM/GLOSSARY/TF-IDF-TERM-FREQUENCY-INVERSE-DOCUMENT-FREQUENCY/](https://WWW.LEARNDATSCI.COM/GLOSSARY/TF-IDF-TERM-FREQUENCY-INVERSE-DOCUMENT-FREQUENCY/)>. ACESSO EM: 8 JUN. 2023.

LEMMATIZATION VS. STEMMING: QUANDO USAR CADA UMA? DISPONÍVEL EM: <[HTTPS://WWW.ALURA.COM.BR/ARTIGOS/LEMMATIZATION-VS-STEMMING-QUANDO-USAR-CADA-UMA?GCLID=CJ0KCQJW98UJBHCGARISAD7QEAIKDNZICJ_ADMRNDAB-3GR_ET6A_YBVBPFDPNP8ZDVTNP5VS9IF8BSAAJQFEALW_WCB](https://WWW.ALURA.COM.BR/ARTIGOS/LEMMATIZATION-VS-STEMMING-QUANDO-USAR-CADA-UMA?GCLID=CJ0KCQJW98UJBHCGARISAD7QEAIKDNZICJ_ADMRNDAB-3GR_ET6A_YBVBPFDPNP8ZDVTNP5VS9IF8BSAAJQFEALW_WCB)>. ACESSO EM: 8 JUN. 2023.



FONTES BIBLIOGRÁFICAS

MADAN, R. TF-IDF/TERM FREQUENCY TECHNIQUE: EASIEST EXPLANATION FOR TEXT CLASSIFICATION IN NLP USING PYTHON (CHATBOT TRAINING ON WORDS). DISPONÍVEL EM: <[HTTPS://MEDIUM.COM/ANALYTICS-VIDHYA/TF-IDF-TERM-FREQUENCY-TECHNIQUE-EASIEST-EXPLANATION-FOR-TEXT-CLASSIFICATION-IN-NLP-WITH-CODE-8CA3912E58C3](https://medium.com/analytics-vidhya/tf-idf-term-frequency-technique-easiest-explanation-for-text-classification-in-nlp-with-code-8ca3912e58c3)>. ACESSO EM: 8 JUN. 2023.

NAULAK, C. A COMPARATIVE STUDY OF NAIVE BAYES CLASSIFIERS WITH IMPROVED TECHNIQUE ON TEXT CLASSIFICATION. 2022. DISPONÍVEL EM: <[HTTPS://WWW.TECHRXIV.ORG/ARTICLES/PREPRINT/A_COMPARATIVE_STUDY_OF_NAIVE_BAYES_CLASSIFIERS_WITH_IMPROVED_TECHNIQUE_ON_TEXT_CLASSIFICATION/19918360/1](https://www.techrxiv.org/articles/preprint/a_comparative_study_of_naive_bayes_classifiers_with_improved_technique_on_text_classification/19918360/1)>. ACESSO EM: 8 JUN. 2023.

NAYAK, P. UNDERSTANDING SEARCHES BETTER THAN EVER BEFORE. DISPONÍVEL EM: <[HTTPS://WWW.BLOG.GOOGLE/PRODUCTS/SEARCH/SEARCH-LANGUAGE-UNDERSTANDING-BERT/](https://www.blog.google/products/search/search-language-understanding-bert/)>. ACESSO EM: 8 JUN. 2023.

OPEN SOURCING BERT: STATE-OF-THE-ART PRE-TRAINING FOR NATURAL LANGUAGE PROCESSING. DISPONÍVEL EM: <[HTTPS://AI.GOOGLEBLOG.COM/2018/11/OPEN-SOURCING-BERT-STATE-OF-ART-PRE.HTML](https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html)>. ACESSO EM: 8 JUN. 2023.

PHATAK, M. SPAM_DETECTION_USING_BERT_MODEL.IPYNB AT MAIN · PHATAK-DEV/TRANSFORMER-MODELS. [S.L: S.N].

RATZ, A. V. MULTINOMIAL NAÏVE BAYES' FOR DOCUMENTS CLASSIFICATION AND NATURAL LANGUAGE PROCESSING (NLP). DISPONÍVEL EM: <[HTTPS://TOWARDSDATASCIENCE.COM/MULTINOMIAL-NA%C3%AFVE-BAYES-FOR-DOCUMENTS-CLASSIFICATION-AND-NATURAL-LANGUAGE-PROCESSING-NLP-E08CC848CE6](https://towardsdatascience.com/multinomial-na%c3%afve-bayes-for-documents-classification-and-natural-language-processing-nlp-e08cc848ce6)>. ACESSO EM: 8 JUN. 2023.

FONTES BIBLIOGRÁFICAS

SACRAMENTO, G. NAIVE BAYES: COMO FUNCIONA ESSE ALGORITMO DE CLASSIFICAÇÃO. DISPONÍVEL EM: <[HTTPS://BLOG.SOMOSTERA.COM/DATA-SCIENCE/NAIVE-BAYES](https://blog.somostera.com/data-science/naive-bayes)>. ACESSO EM: 8 JUN. 2023.

SANH, V. SMALLER, FASTER, CHEAPER, LIGHTER: INTRODUCING DISTILBERT, A DISTILLED VERSION OF BERT. DISPONÍVEL EM: <[HTTPS://MEDIUM.COM/HUGGINGFACE/DISTILBERT-8CF3380435B5](https://medium.com/huggingface/distilbert-8cf3380435b5)>. ACESSO EM: 8 JUN. 2023.

SCHUMACHER, NÍCOLAS MACHADO. ESTUDO COMPARATIVO DE MODELOS DE APRENDIZADO DE MÁQUINA PARA DETECCÃO DE EMAIL SPAM. 2020. 78 F., IL. TRABALHO DE CONCLUSÃO DE CURSO (BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO)—UNIVERSIDADE DE BRASÍLIA, BRASÍLIA, 2020.

SHUKLA, P. PERFORMING EMAIL SPAM DETECTION USING BERT IN PYTHON. DISPONÍVEL EM: <[HTTP://WWW.ANALYTICSVIDHYA.COM/BLOG/2021/09/PERFORMING-EMAIL-SPAM-DETECTION-USING-BERT-IN-PYTHON/](http://www.analyticsvidhya.com/blog/2021/09/performing-email-spam-detection-using-bert-in-python/)>. ACESSO EM: 8 JUN. 2023.

TRANSFORMER: A NOVEL NEURAL NETWORK ARCHITECTURE FOR LANGUAGE UNDERSTANDING. DISPONÍVEL EM: <[HTTPS://AI.GOOGLEBLOG.COM/2017/08/TRANSFORMER-NOVEL-NEURAL-NETWORK.HTML](https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html)>. ACESSO EM: 8 JUN. 2023.

VIDANYA, B. PHISHING COM O NOME DA HOSTINGER: COMO IDENTIFICAR E SE PREVENIR. DISPONÍVEL EM: <[HTTP://WWW.HOSTINGER.COM.BR/BLOG/CUIDADO-COM-O-EMAIL-DE-PHISHING/](http://www.hostinger.com.br/blog/cuidado-com-o-email-de-phishing/)>. ACESSO EM: 8 JUN. 2023.