USO DE PROCESSAMENTO DE LINGUAGEM NATURAL PARA CLASSIFICAÇÃO DE *TWEETS*

Amanda Isabela de Campos[1]

[1]COPPE, Programa de Engenharia Civil, Universidade Federal do Rio de Janeiro

amanda.campos@coc.ufrj.br

Resumo: No presente trabalho são comparados os desempenhos de modelos de classificação que adotam o processamento de linguagem natural e aprendizado de máquina (*machine learning*) para prever se determinado *tweet* (frases que usuários da rede social *Twitter* publicam) trata sobre um desastre ou não, dessa forma autoridades responsáveis conseguem mapear emergências a partir de redes sociais. Os modelos de regressão logística e rede neural convolucional (CNN) foram aplicados para classificação onde a CNN obteve melhor desempenho, ou seja, consegue prever melhor a qual classe o *tweet* pertence.

1. Introdução

O processamento de linguagem natural (NLP) tem como principais aplicações extrair significados e aprender a partir de dados de textos. Segundo Hirschberg e Manning [1] o NLP emprega técnicas computacionais com o propósito de aprender, compreender e produzir conteúdo a partir da linguagem humana. As primeiras abordagens computacionais da pesquisa linguística concentraram-se na automatização e análise de estruturas e no desenvolvimento de tecnologias básicas, como tradução automática e reconhecimento da fala. Os pesquisadores atuais aplicam essas ferramentas para a criação de sistemas de diálogo automático, motores de tradução, mineração de mídias sociais com informações sobre saúde ou finanças, e identificação sentimentos e emoções.

As redes sociais podem ser um forte aliado das autoridades no sentido de informar desastres ou acidentes, porém a classificação automática deve ser precisa, ou seja, se o texto publicado pelo usuário trata de um desastre ou não. Para demonstrar isso, neste trabalho é analisado um conjunto de dados constituído de mais de 10.000 *tweets* (frases publicadas por usuários do Twitter) com palavras

que indicam acidentes como "fogo", "quarentena" e "pandemônio". O conjunto de dados é denominado *Disasters on Social Media* e está disponível livremente no site DataWorld [2].

O presente trabalho tem como objetivo aplicar algoritmos de inteligência artificial com métodos de processamento de linguagem natural (NLP) em um conjunto de dados denominado *Disasters on Social Media*, dessa forma é possível prever ou classificar, quais *tweets* tratam sobre um evento de catástrofe e quais tratam apenas de uma questão irrelevante com objetivo de criar uma aplicação que notifique as autoridades responsáveis em caso de emergências. O problema é que em alguns casos as mesmas palavras são usadas para tratar de um filme por exemplo, portanto, o modelo deve ser capaz de detectar diferenças mínimas.

2. Tecnologia

A análise e caracterização dos dados, bem como a aplicação dos modelos de inteligência artificial serão implementadas com a linguagem de programação Python, por ser: (i) uma linguagem de programação simples, livre e aberta; (ii) a linguagem mais usada atualmente e (iii) composta de várias bibliotecas desenvolvidas e em constante atualização já implementadas para aplicações de inteligência artificial e processamento de linguagem natural. No presente trabalho adotou-se as bibliotecas Numpy [3] (para cálculos numéricos e operações com matrizes), Pandas [4] (para manipulação de dataset em formato de tabelas e planilhas), Matplotlib [5] (para a geração de gráficos e visualização dos dados), Seaborn [6] (também para a geração de gráficos, baseado no matplotlib porém mais voltado para estatística), ScikitLearn [7] (biblioteca de inteligência artificial, com modelos já implementados de classificação, regressão, etc) e Spark NLP [8] (biblioteca para processamento avançado de linguagem natural).

3. Metodologia:

3.1 Apresentação dos pré-processamentos realizados

A etapa de pré-processamento tem elevada importância para a qualidade dos resultados em problemas de NLP. Segundo Symeonidis et al. [9] este é o primeiro passo na classificação de textos e a escolha de técnicas corretas de pré-processamento pode melhorar a eficácia da classificação. Este indica que técnicas como a lemmatização, remoção de números e substituição de contrações, melhoram a precisão, enquanto outras como a remoção de pontuação não.

Neste trabalho a primeira fase do pré-processamento consiste em remover todos os caracteres irrelevantes, como quaisquer caracteres não alfanuméricos que não acrescentam informações

importantes, como por exemplo "@" e urls. Transformar os caracteres maiúsculos em minúsculos, uma vez que essa diferença não adiciona nenhum conteúdo e devem ser tratadas da mesma forma. Além de considerar a combinação de palavras mal escritas com uma única representação. E por fim realizar a lemmatização, ou seja, reduzir as palavras ao radical e não apresentar todas as variantes de cada palavra.

O próximo passo é realizar a *tokenização* do texto, ou seja, separar as frases em palavras individuais. Para isso, será adotada a biblioteca *RegexpTokenizer* do *nltk.tokenize* existente no Python. Além disso, será aplicada a técnica de *One-hot-encoding* com o modelo *Bag of Words* para representar palavras em forma de números, adicionando um índice a cada palavra do vocabulário, de forma que seja irrelevante a ordem que as palavras estão localizadas na frase. Por fim as palavras serão transformadas em vetores e conjunto de dados será dividido entre treino e teste na proporção de 80 e 20%.

3.2 Descrição matemática dos modelos empregados

- **Regressão Logística**

O modelo de regressão logística é um modelo de classificação linear utilizado para estivar a probabilidade de uma instância pertencer a uma determinada classe [10]. Se esta probabilidade forma maior que 50% então o modelo prevê que a instância pertence à classe positiva (rotulada como 1), caso contrário o modelo prevê que pertence à classe negativa (rotulada como 0). No modelo de regressão logística a probabilidade estimada é calculada com a Eq. 1. E a logística, que transforma um número entre 0 e 1, é uma função sigmoide apresentada na Eq. 2.

$$\hat{p} = h_\theta(\boldsymbol{x}) = \sigma(\theta^T . \boldsymbol{x}) \tag{1}$$

$$\sigma(t) = \frac{1}{1+\exp(-t)} \tag{2}$$

A previsão do modelo é realizada com a Eq. 3, uma vez que tem-se a probabilidade estimada $\hat{p} = h_\theta(\boldsymbol{x})$ que a instancia pertence ou não a classe positiva.

$$\hat{y} = \begin{cases} 0 \; se \; \hat{p} < 0.5 \\ 1 \; se \; \hat{p} \geq 0.5 \end{cases} \tag{3}$$

O modelo de regressão logística é treinado de forma a definir o vetor do parâmetro θ para o modelo estimar altas probabilidades para instâncias positivas e baixas probabilidades para instâncias

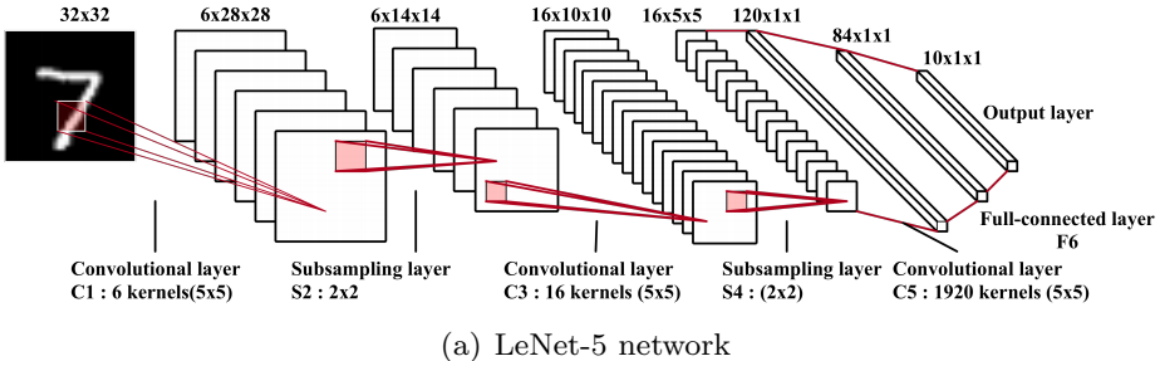
negativas (Eq. 4). E por fim essa função é minimizada com o Método do gradiente descendente para encontrar o mínimo global.

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\left(\hat{p}^{(i)}\right) + \left(1 - y^{(i)}\right)\log\left(1 - \hat{p}^{(i)}\right)\right] \qquad (4)$$

- **Redes Neurais Convolucionais (CNN)**

As redes neurais convolucionais, em inglês, *convolutional neural network* ou CNN, foram primeiramente adotadas em problemas de visão computacional, baseado em aprendizado profundo de redes neurais. A figura 1 mostra a arquitetura da LeNet-5 (Lecun et al., 1998) para reconhecimento de documentos.

Segundo Guo *et al*. [11] uma rede neural convolucional inclui principalmente três tipos de camadas, são elas, camada convolucional, camada de pooling e camada totalmente conectada. De acordo com Gu *et al*. [12] a classificação de texto é uma tarefa crucial para o Processamento de Linguagem Natural (PNL) e como as frases em linguagem natural têm estruturas complicadas, tanto sequenciais como hierárquicas, que são essenciais para entendê-las, as CNN possuem à poderosa capacidade de capturar as relações locais das estruturas temporais ou hierárquicas e alcançaram o melhor desempenho na modelagem de sentenças. Estes ainda indicam que uma arquitetura CNN adequada é importante para a classificação de textos.



(a) LeNet-5 network

Figura 1. Arquitetura de CNN. Fonte: Falcão et al., 2013

3.3 Análise dos resultados

A matriz de confusão é o método mais indicado para avaliar o desempenho de um modelo de classificação [10]. Em problemas de duas classes a matriz é construída da seguinte forma: cada linha em uma matriz de confusão representa uma classe real e cada coluna representa uma classe prevista. Neste caso, a matriz será 2x2 (Fig. 2) e a primeira posição representa o número de verdadeiro

positivos, ou seja, valores que são verdadeiros e foram classificados como tal, porém a posição primeira linha e segunda coluna representa o número de falsos positivos, valores que correspondem a classe negativo e foram classificados como positivos, e assim respectivamente, a posição da segunda linha e primeira coluna corresponde aos falsos negativos, ou seja, pertencem a classe.

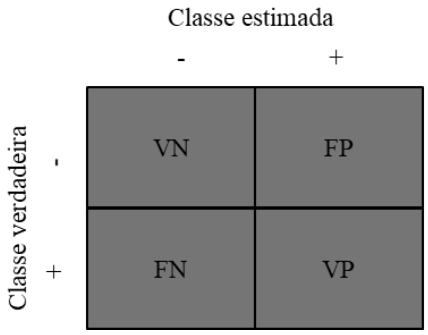


Figura 2. Matriz de confusão

Da matriz de confusão podem ser retiradas outras métricas como a acurácia das previsões positivas, também chamada de precisão do classificador, indicada na Eq. 5. Onde *VP* é o número de verdadeiros positivos e FP é o número de falsos positivos. A precisão é analisada em conjunto métrica revocação (Eq. 6), que pode ser entendida como a sensibilidade ou taxa de verdadeiros positivos, ou seja é a taxa de registros positivos que são corretamente classificadas. Onde *FN* é o número de falsos negativos. Na biblioteca *scikit learn* essas métricas são calculadas com a função precision_score (precisão) e recall_score (revocação).

$$precisão = \frac{VP}{VP+FP} \tag{5}$$

$$revocação = \frac{VP}{VP+FN} \tag{6}$$

Por fim, é comum combinar precisão e revocação em um único incide, chamado de pontuação F_1, que é a média harmônica entre as duas métricas apresentadas anteriormente (Eq. 7). No *scikit learn* essa métrica é calculada com o comando F1_score.

$$F_1 = \frac{2}{\frac{1}{precisão}+\frac{1}{revocação}} = \frac{VP}{VP+\frac{FN+FP}{2}} \tag{7}$$

Além do cálculo das métricas, a avaliação dos resultados após a aplicação dos modelos de classificação será realizada a partir da decomposição PCA, que projeta os dados em duas dimensões, dessa forma é possível prever se o problema é ou não separável e para visualizar se os embeddings estão capturando informações relevantes para o nosso problema. Serão também avaliadas quais

palavras estão sendo usadas para realizar as previsões, ou seja, a importância que cada palavra teve para o modelo de classificação.

A pontuação TF-IDF (Frequência do termo, Frequência inversa do documento) será adicionada no modelo *Bag of Words* para auxiliar o modelo a encontrar as palavras mais significativas que tem maior peso.

4. Resultados

4.1 Pré-processamento

A análise exploratória do conjunto de dados indicou que este é composto por mais de 10.000 frases e que todos possuem palavras como "fogo", "quarentena" e "pandemônio", as Fig. 3 e Fig. 4 indicam respectivamente os registros nas cinco primeiras e cinco últimas linhas do conjunto de dados, onde observa-se exemplos de frases classificadas como “relevant” ou seja, correspondem a desastres. Estes foram classificados se estavam se referindo a um desastre natural ou relacionados a uma piada, critica de filme ou outro evento não referente a uma calamidade. Sendo, portanto, um problema de classificação de três classes: “not relevante”, “relevant” e “can't decide”. O número de registros em cada classe é respectivamente 6187, 4673 e 16 dados, como apontado na Fig. 5.

	text	choose_one	class_label
0	Just happened a terrible car crash	Relevant	1
1	Our Deeds are the Reason of this #earthquake M...	Relevant	1
2	Heard about #earthquake is different cities, s...	Relevant	1
3	there is a forest fire at spot pond, geese are...	Relevant	1
4	Forest fire near La Ronge Sask. Canada	Relevant	1

Figura 3. Primeiras linhas do conjunto de dados

	text	choose_one	class_label
10854	1.3 #Earthquake in 9Km Ssw Of Anza California ...	Relevant	1
10855	Evacuation order lifted for town of Roosevelt:...	Relevant	1
10856	See the 16yr old PKK suicide bomber who detona...	Relevant	1
10857	To conference attendees! The blue line from th...	Relevant	1
10858	The death toll in a #IS-suicide car bombing on...	Relevant	1

Figura 4. Últimas linhas do conjunto de dados

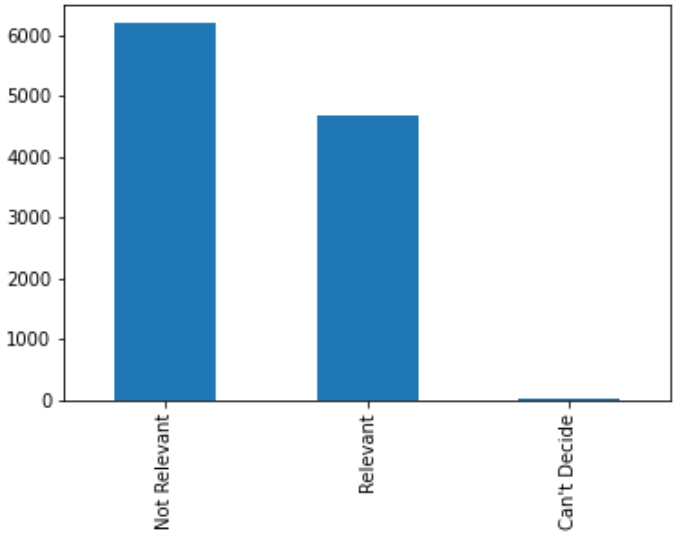


Figure 5. Descrição das variáveis

A Fig. 6 indica as primeiras linhas do conjunto de dados depois da retirada dos caracteres irrelevantes.

	text	choose_one	class_label
0	just happened a terrible car crash	Relevant	1
1	our deeds are the reason of this earthquake m...	Relevant	1
2	heard about earthquake is different cities, s...	Relevant	1
3	there is a forest fire at spot pond, geese are...	Relevant	1
4	forest fire near la ronge sask canada	Relevant	1

Figura 6. Primeiras linhas do conjunto de dados depois da remoção de caracteres irrelevantes

A Fig. 7 apresenta os tokens das primeiras linhas do conjunto de dados após aplicar este tokenização.

	Unnamed: 0	text	choose_one	class_label	tokens
0	0	just happened a terrible car crash	Relevant	1	[just, happened, a, terrible, car, crash]
1	1	our deeds are the reason of this earthquake m...	Relevant	1	[our, deeds, are, the, reason, of, this, earth...
2	2	heard about earthquake is different cities, s...	Relevant	1	[heard, about, earthquake, is, different, citi...
3	3	there is a forest fire at spot pond, geese are...	Relevant	1	[there, is, a, forest, fire, at, spot, pond, g...
4	4	forest fire near la ronge sask canada	Relevant	1	[forest, fire, near, la, ronge, sask, canada]

Figura 7. Tokens do conjunto de dados

Após esse processo observou-se que existem 154724 palavras no total, com um tamanho de vocabulário de 18101. A Fig. 8 apresenta o histograma do comprimento das frases, observa-se que o comprimento máximo da sentença é de 34.

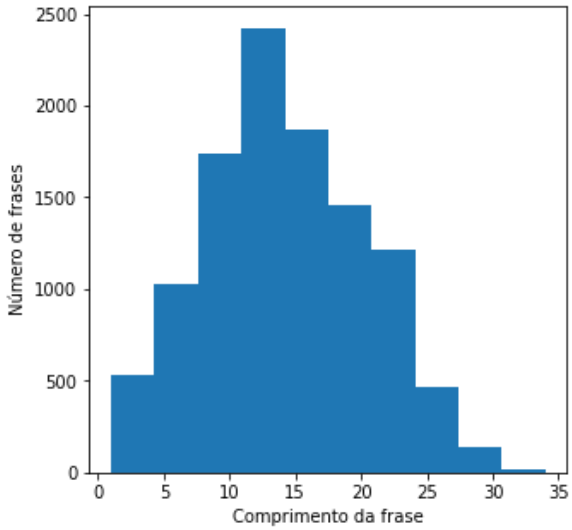


Figura 8. Histograma do comprimento de frases

4.2 Classificação

A Fig. 9 apresenta a decomposição PCA, onde é possível prever que o problema é não separável.

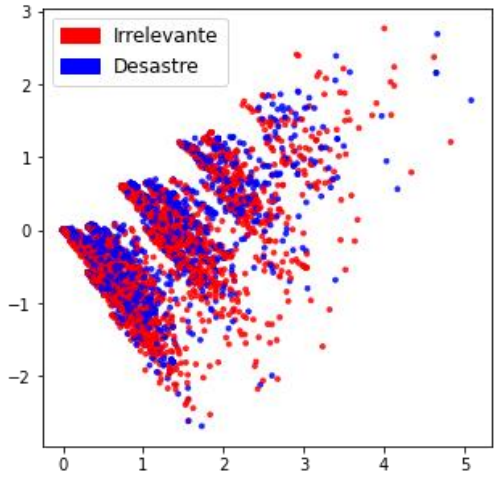


Figura 9. Decomposição PCA

- **Regressão Logistica**

O modelo de regressão logística foi adotado com o hiperparametro C=30.0 e por se tratar de um problema com classes desbalanceadas foi aplicado o comando: class_weight = 'balanced' para que o modelo contorne o problema de desbalanceamento. Como resultado este modelo obteve acurácia = 0.754, precisão = 0.752, recall = 0.754, f1 = 0.753 e na Fig. 10 está ilustrada a matriz de confusão resultante.

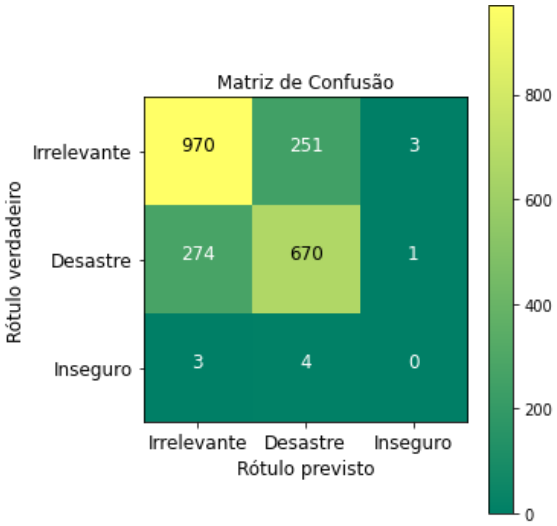


Figura 10. Matriz de confusão

A Fig. 11 apresenta a importância de cada palavra que o modelo de classificação regressão logística adotou.

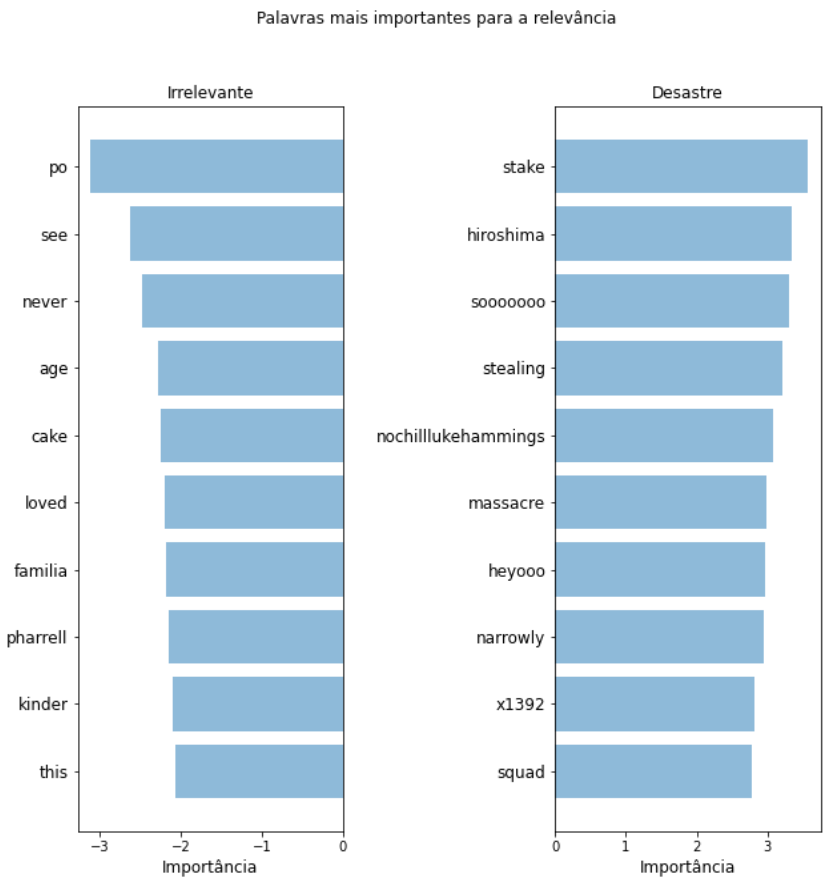


Figura 11. Importância de cada palavra para o modelo de classificação

Como estratégia para melhorar o acurácia do modelo na etapa de *Bag-of-Words* será adotado um TF-IDF (*Term Frequency, Inverse Document Frequency*) que aplica peso as palavras de acordo com a frequência que estas aparecem no conjunto de dados, ou seja, palavras pouco frequentes não terão muita influência pois só causam ruídos ao modelo. Todas as análises serão refeitas com essa estratégia. A Fig. 12 apresenta a decomposição PCA que neste caso tem um aspecto de problema mais separável o que provavelmente resultara em um melhor desempenho do modelo.

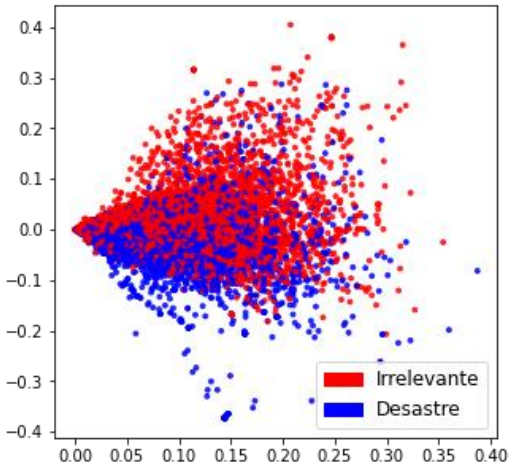


Figura 12. Decomposição PCA

Como resultados foi obtido: accuracy = 0.762, precision = 0.760, recall = 0.762, f1 = 0.761 que são valores maiores que os obtidos anteriormente o que indica que a inclusão do TF-IDF (*Term Frequency, Inverse Document Frequency*) resultou em melhoras ao modelo. A Fig. 13 ilustra a matriz de confusão para este caso, onde observa-se que o número de falsos positivos diminuiu.

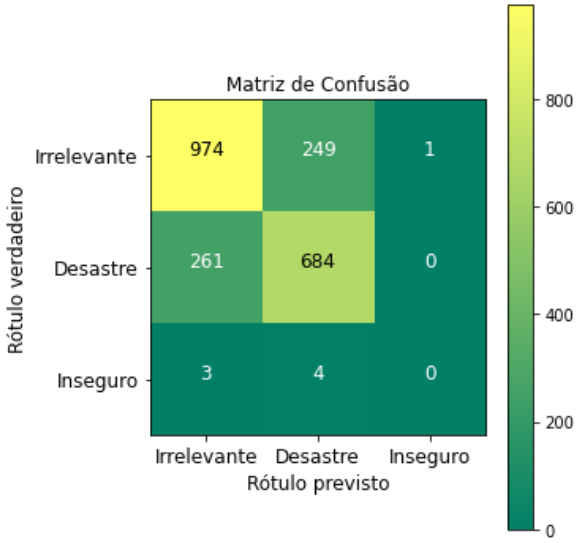


Figura 13. Matriz de Confusão

A Fig. 14 apresenta a importância das palavras para o modelo, neste caso, observa-se que as palavras apresentaram um índice de importância muito maior depois da modificação, o que era esperado.

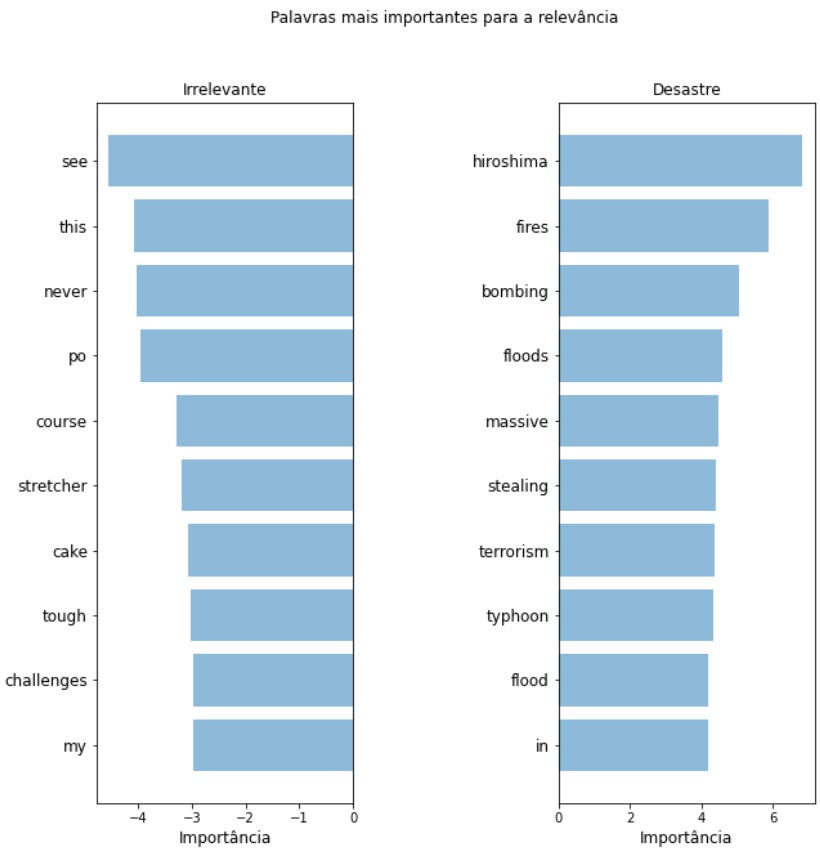


Figura 14. Importância de cada palavra para o modelo de classificação

- **Rede Neural Convolucional**

A Fig. 15 apresenta a arquitetura da rede neural convolucional adotada neste trabalho, o otimizador Adam foi utilizado e por se tratar de um problema de classificação multiclasse a loss indicado é do tipo *'binary_crossentropy'*, o modelo será treinado por 10 épocas e a dimensão do embedding igual a 50.

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_2 (Embedding)      (None, 140, 50)           500000
_________________________________________________________________
flatten_2 (Flatten)          (None, 7000)              0
_________________________________________________________________
dense_3 (Dense)              (None, 32)                224032
_________________________________________________________________
dense_4 (Dense)              (None, 1)                 33
=================================================================
Total params: 724,065
Trainable params: 724,065
Non-trainable params: 0
```

Figura 15. Arquitetura da rede neural

Como resultado a acurácia obtida foi de accuracy = 0.788, precision = 0.787, recall = 0.788, f1 = 0.786. A Fig. 16 apresenta a matriz de confusão do modelo de rede neural convolucional.

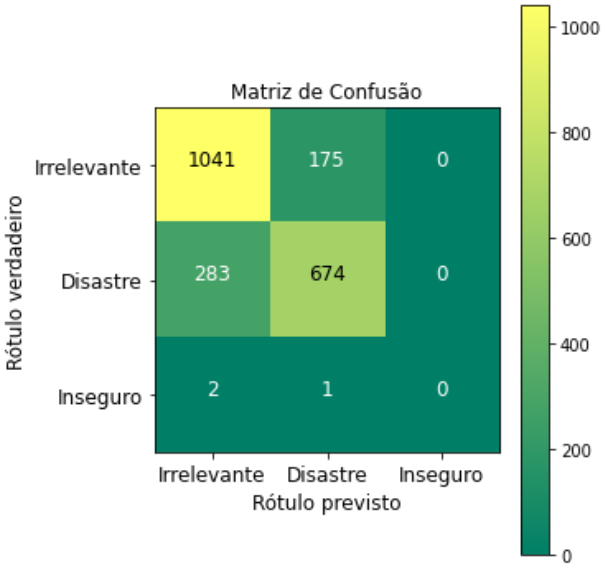


Figura 16. Matriz de Confusão do modelo de rede neural convolucional

5. Conclusões

O presente trabalho conseguiu, através de modelos de processamento de linguagem natural e aprendizado de máquinas, classificar se twetts de um conjunto dados denominado *Disasters on Social Media*, estão tratando sobre um desastre ou sobre outro assunto qualquer, com até 79% de acurácia.

Uma publicação em redes social referente a um acidente pode ter muitas palavras parecidas do que um referente a um comentário sobre um filme de ação, por isso o modelo de classificação deve ser potente no sentido de distinguir as diferenças mais sutis do texto.

No presente trabalho dedicou-se uma certa cautela à etapa de pré-processamento com lemmatização, tokenização, remoção de números e caracteres irrelevantes e substituição de contrações, acredita-se que por isso os resultados dos modelos foram satisfatórios. Pode-se concluir que o pré-processamento tem papel fundamental na qualidade final dos resultados. Além disso, o cuidado com o tratamento do conjunto de dados desbalanceados, como o treinamento dos modelos de forma que esse desbalanceamento seja convertido, resultou em resultados mais confiáveis.

Com a execução de todos os modelos de classificação propostos neste trabalho pode-se afirmar que o modelo mais indicado para este conjunto de dados é o modelo de rede neural convolucional porque este foi o modelo que apresentou maior acurácia e matriz de confusão com

mais valores na diagonal princial, além das métricas precisão, recall e F1-score com valores mais próximos da unidade. Este modelo de classificação ainda poderia ser melhorado com estudos na busca dos parâmetros com a técnica de *Grid Search*, ou teste de diferentes arquiteturas, como por exemplo aplicação de modelos pré-treinados, essa recomendação é dada para trabalhos futuros.

Referências

[1] HIRSCHBERG, Julia; MANNING, Christopher D. Advances in natural language processing. Science, v. 349, n. 6245, p. 261-266, 2015.

[2] Dataset Disasters on Social Media. Data.World. Disponível em: https://data.world/crowdflower/disasters-on-social-media. Acesso em: 15, dezembro de 2020.

[3] NumPy v1.19 Manual Documentation. Disponível em: <https://numpy.org/doc/stable/reference/>. Acesso em: 17 de ago. de 2020.

[4] Pandas v1.11 Documentation. Disponível em: <https://pandas.pydata.org/docs/>. Acesso em: 17 de ago. de 2020.

[5] Matplotlib: Visualization with Python v3.3.1. Disponível em: <https://matplotlib.org/contents.html>. Acesso em: 17 de ago. de 2020.

[6] Seaborn: Statistical data visualization. Disponível em: <https://seaborn.pydata.org/tutorial.html>. Acesso em: 17 de ago. de 2020.

[7] ScikitLearn: Machine Learning in Python v0.23.2. Disponível em: <https://scikit-learn.org/stable/user_guide.html>. Acesso em: 18 de ago. de 2020.

[8] THOMAS, Alex. Natural Language Processing with Spark NLP: Learning to Understand Text at Scale. Editora O'Reilly Media, Incorporated, 2020.

[9] SYMEONIDIS, Symeon; EFFROSYNIDIS, Dimitrios; ARAMPATZIS, Avi. A comparative evaluation of pre-processing techniques and their interactions for twitter sentiment analysis. Expert Systems with Applications, v. 110, p. 298-310, 2018.

[10] Géron, A. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: LecunY, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11):2278-2324.

[11] T. Guo, J. Dong, H. Li and Y. Gao, "Simple convolutional neural network on image classification," 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA), Beijing, 2017, pp. 721-724, doi: 10.1109/ICBDA.2017.8078730.

[12] GU, Jiuxiang et al. Recent advances in convolutional neural networks. Pattern Recognition, v. 77, p. 354-377, 2018.