UNIVERSIDADE FEDERAL DO PARANÁ

Leandro Duarte Pulgatti

RELATÓRIO DE TRABALHO FINAL DE DISCIPLINA

CI1030-ERE2-CIÊNCIADEDADOS

DETECÇÃO DE SPAM EM E-MAIL UTILIZANDO NLP

Curitiba, PR
2021

1 ANÁLISE DA BASE DE DADOS

O dataset escolhido é uma compilação de e-mails da empesa 'Enron' nos anos de 2005 e 2006 e foi categorizado como Spam ou Normal pelos próprios usuários.

Estes dados foram compilados a partir das informações retiradas de

http://www2.aueb.gr/users/ion/data/enron-spam/ e foram primeiro utilizados e descritos em (METSIS; ANDROUTSOPOULOS; PALIOURAS, 2006)[1].

O arquivo de composto de 4 campos

- ID - Número sequencial
- label - Atributo alfabético indicando a classe a qual o e-mail pertence

  ham - email normal

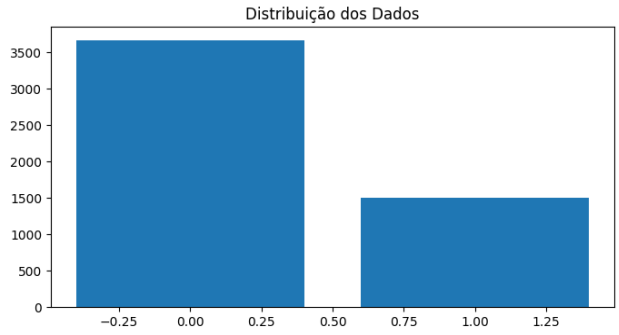
  spam - email classificado como spam
- text - o texto completo do e-mail, incluindo subject e o corpo do mesmo
- label_num - Atributo numérico indicando a classe a qual o e-mail pertence

  0 - email normal

  1 - email classificado como spam

O dataset possui um total de 5171 linhas sendo 3672 (71.01%) classificadas como Ham (normal) e 1499 (28.99%) classificadas como Spam.

Figura 1.1 – Distribuição do campo LABEL na base de dados



Fonte: Criado pelo Autor.

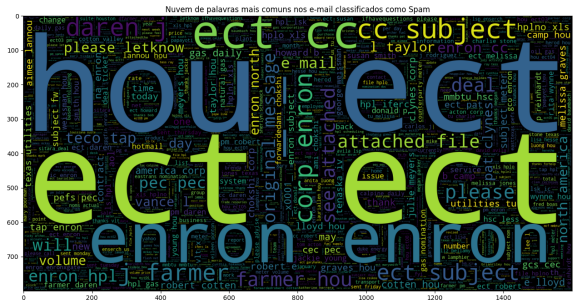
A amostra está desbalanceada, porem reflete uma visão do mundo real para o momento do corte executado.

Como a primeira coluna (ID) é apenas um sequencial e não influi no resultado final, ela foi retirada do dataset a ser processado.

A coluna 'text' possui o corpo do e-mail. Foram restirados os campo que indicam os e-mails envolvidos, tando do remetente quanto dos receptores. Esta é a coluna que contêm os dados que serão realmente utilizados no treinamento. Um total de 88263 palavras estão presentes no campo 'text'.

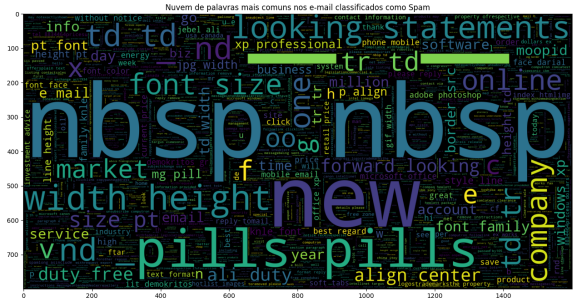
[1] Os dados originais podem ser baixados em https://www.kaggle.com/venky73/spam-mails-dataset?select=spam_ham_dataset.csv

Figura 1.2 – Nuvem de palavras fraquentes em e-mails identificados como normal



Fonte: Criado pelo Autor.

Figura 1.3 – Nuvem de palavras fraquentes em e-mails identificados como Spam



Fonte: Criado pelo Autor.

O campo 'text' não está processado, desta maneira o texto original possui uma série de erros e abreviações, bem como recursos linguísticos que dificultam o processamento utilizando técnicas como 'word embeddings', visto que estas necessitam de palavras conhecidas e previamente treinadas para que se possa capturar seu significado.

Excerto das 5 primeiras linhas do campo 'text' sem pré-processamento

```
0  Subject: enron methanol ; meter # : 988291\nth...
1  Subject: hpl nom for january 9 , 2001\n( see a...
2  Subject: neon retreat\nho ho ho , we ' re arou...
3  Subject: photoshop , windows , office . cheap ...
4  Subject: re : indian springs\nthis deal is to ...
```

Os dados são então tratados retirando as 'stop words'. Os 'stop words' são palavras que são comuns e não agregam na classificação, para este trabalho foi utilizada lista, em inglês, da ferramenta 'nltk' (TOOLKIT, 2021).

O termo 'Subject:' aparece em todos as entradas, assim não contribui para a classificação, por isto foi retirado também.

Excerto das 5 primeiras linhas do campo 'text' após o pré-processamento

```
0   enron methanol ; meter   : 988291isfollowtono...
1   hpl nomjanuary 9 , 2001 ( see attached file :...
```

```
2   neon retreat ho ho ho ,'around tomost wonderf...
3   photoshop , windows , office . cheap . main t...
4  : indian springsdealto bookteco pvr revenue .i...
```

2 TREINAMENTO E TESTE

Devido a questões já colocadas na análise do texto, com a presença de palavras não comuns na língua inglesa, imprecisão de termos e erro de escrita, foi escolhido a utilização de uma variação das Rede Neurais Recorrentes (RNR) chamada LSTM (LSTM - Long short-term memory) para a classificação dos e-mails.

Diferente de um Rede neural convolucional que 'esquece' os dados originais após cada iteração, uma RNR utiliza destes dados, mais os novos dados gerados para o próximo processamento. Isto permite que mesma 'lembre' do contexto original do termo dentro de um texto.

As rede LSTM foram criadas para evitar que os gradientes que são retro-propagados não 'desapareçam' (tendam a zero). Estas redes podem processar sequências de dados maiores que as RNR tradicionais.

Este tipo de rede neural é muito utilizada em tarefas de processamento de linguagem natural pois capturam melhor a semântica das palavras. Ou seja, capturam o contexto no qual a palavra está inserida que pode influenciar no seu significado.

Os e-mails foram aleatoriamente separados em dois grupos, Treinamento e Teste. A primeira distribuição testada foi de 80% para treinamento e os outros 20% para teste.

Os dados ficaram assim distribuídos:

- Tamanho Base Treino: 4136
- Tamanho Base Teste 1035

Outras distribuições foram testas e os melhores resultados foram obtidos com uma divisão de 60/40 para Treino e Teste. Este resultado, provavelmente, se deve ao fato de uma distribuição mais próxima de 50% captura melhor a distribuição normal entre dados com os diversos rótulos. Uma distribuição 80/20 pode conter uma distribuição muito desbalanceada, com poucos exemplos de Spam na base de treinamento. Uma discussão sobre este problema e como mitigar é realizada na seção 3.

Para este trabalho os dados serão tratado como eles se apresentam, e os valores finais serão calculados com uma média de várias execuções para minimizar estes problemas.

A maioria dos hiper parâmetros da rede fora mantidos com seus valores padrão. Os parâmetros alterados foram :

- n_lstm = 200

  Número de camadas da rede.

- drop_lstm = 0.2

  para a execução da rede quando o ganho fica inferior a um percentual a cada iteração. Ajuda a diminuir o 'overfitting'.

- embeding_dim = 16

  Número de dimensões do array.

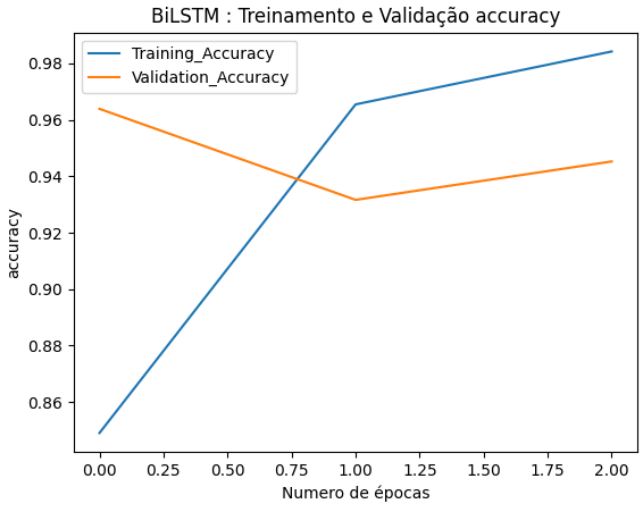
Os experimentos foram executados com 30 épocas, entretanto devido ao parâmetro 'dropout' a maioria das execução terminou sempre entre a 3 e a 4 execução. Como a acurácia e a perda jé inciam em valores altos, acurácia acima de 80%, fica fácil para o modelo convergir rapidamente.

Excerto da tela de uma das execuções mostrando a evolução do modelo no treinamento

```
Epoch 1/30
130/130 - 33s - loss: 0.3686 - accuracy: 0.8374 - val_loss: 0.0994 -
val_accuracy: 0.9712
Epoch 2/30
130/130 - 25s - loss: 0.0436 - accuracy: 0.9893 - val_loss: 0.1239 -
val_accuracy: 0.9671
Epoch 3/30
130/130 - 25s - loss: 0.0147 - accuracy: 0.9976 - val_loss: 0.1554 -
val_accuracy: 0.9539
Resultado do teste - Perda: 0.15537875890731812 - Acurácia: 95.3855037689209%
```

	Training_Loss	Training_Accuracy	Validation_Loss	Validation_Accuracy
0	0.368592	0.837442	0.099365	0.971208
1	0.043622	0.989294	0.123909	0.967130
2	0.014737	0.997601	0.155379	0.953855

Figura 2.1 – Acurácia obtida durante as épocas de treinamento



Fonte: Criado pelo Autor.

3 ANÁLISE E CONCLUSÃO

Nos experimentos a média obtida foi de 98% de Acurácia na base de treinamento e de 93% na base de testes

	precision	recall	f1-score	support
0	0.99	0.99	0.99	742
1	0.98	0.98	0.98	293
accuracy			0.99	1035
macro avg	0.99	0.99	0.99	1035
weighted avg	0.99	0.99	0.99	1035

Estes valores, apesar de significativos, podem indicar que o desbalanceamento da base pode estar gerando um resultado difícil de se reproduzir no mundo real. Assim se o modelo apenas escolhesse todos os rótulos como sendo 'Ham' ele já acertaria 70% das vezes.

Uma análise da matriz de confusão indica que, proporcionalmente, os dados classificados como 'Spam' possuem um erro maior do que os classificados como normal.

Ham	Spam
736	6
5	288

Outra questão é que, como a base provém de apenas uma empresa, os dados normais tendem a conter mais palavras que pertencem ao cotidiano de negócios daquela empresa, assim um e-mail sem nenhuma, ou poucas, palavra que remetam ao negócio, pode facilmente ser colocado em uma outra categoria.

Ambas as questões levam a desconfiança de que o modelo está sofrendo de 'overfitting', ou seja ele está treinado não para reconhecer dados de spam gerais, mas apenas dados daquela empresa.

Para se remediar estas questões é algumas técnicas são possíveis:

- 'Downsampling' - gerando uma base com o mesmo número de entrada de cada rótulo.

  Esta técnica apresenta como problema o fato de que pode distorcer a análise para o outro lado, ou seja o label que foi diminuído pode ficar mal representado

- 'Dados Sintéticos' - gerando dados adicionais na base para aumentar o número de ocorrência de um determinado label.

  Neste caso é complexa a ageração de dados sintéticos, pois não se tem como conseguir, a partir dos dados existentes, a criação de novas combinações 'válidas'. Isto aumentaria ainda mais o viés e o 'overfitting'.

- 'Novas bases' - Agregar uma base maior

  Com mais exemplos e uma representatividade mais abrangente de assuntos é a melhor solução para este caso.

Este trabalho teve por finalidade mostrar o processo completo de uma análise de dados para reconhecimento de Spam em e-mails. Ficou claro que a escolha da base de dados possui influência significativa no resultado final.

Os passos realizados e descritos neste relatório estão distribuídos da seguinte maneira

1. Coleta e análise dos dados
2. Pré-processamento
3. Escolha do melhor algorítimo
4. Análise dos resultados
5. Possibilidades de melhora

REFERÊNCIAS BIBLIOGRÁFICAS

METSIS, V.; ANDROUTSOPOULOS, I.; PALIOURAS, G. Spam filtering with naive bayes-which naive bayes? In: MOUNTAIN VIEW, CA. **CEAS**. [S.l.], 2006. v. 17, p. 28–69.

TOOLKIT, N. L. **Natural Language Toolkit**. 2021. <https://www.nltk.org/>.