

MPEI Projeto 1

Gonçalo Ribau (119560), Filipe Marques (120303)

December 2024

1 Introdução

Neste trabalho construímos um programa capaz de analisar notícias e classificá-las como verdadeiras ou falsas de acordo com um Dataset pré definido. Escolhemos este tema porque achámos interessante tentarmos aplicar os conhecimentos adquiridos nas aulas a esta situação do dia-a-dia e também pareceu um bom desafio. Utilizámos o Bloom Filter, o Classificador de Naive Bayes e o Minhash em conjunto de forma a atingir este objetivo.

2 Aplicação Conjunta

A nossa aplicação tem como objetivo principal identificar se uma notícia é falsa. O programa começa por solicitar a fonte da notícia ao utilizador. Utilizando um Bloom Filter, que "armazena" fontes associadas a notícias falsas, o sistema verifica se a fonte fornecida é considerada não credível. Caso a fonte seja identificada como não confiável, o programa termina imediatamente e informa que a notícia não é legítima.

Se a fonte não estiver associada às fontes no filtro, o programa prossegue, solicitando o título e o conteúdo da notícia. Com base nesses dados, o sistema utiliza o Classificador Naive Bayes para determinar se a notícia é verdadeira ou falsa.

O papel do MinHash neste trabalho é simples, mas fundamental. Este módulo é responsável por otimizar o banco de notícias de treino utilizado pelo classificador. Dado que trabalhamos com um volume significativo de notícias, tanto verdadeiras quanto falsas, a probabilidade de existirem notícias semelhantes abordando o mesmo tema é elevada. Esse cenário pode ser problemático, pois, ao treinar o classificador com notícias "repetidas", mesmo que apresentem pequenas diferenças, acabamos por atribuir uma importância excessiva a conteúdos redundantes. Isso pode prejudicar o desempenho e a precisão dos resultados finais. O uso do MinHash ajuda a identificar e remover estas duplicações, garantindo um treino mais eficiente e equilibrado do classificador.

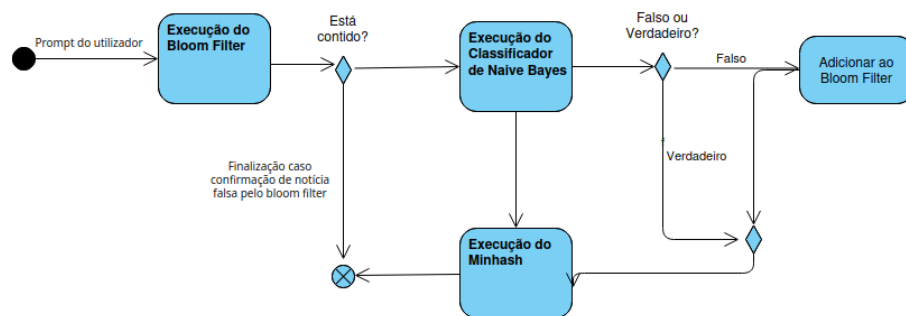


Figure 1: Flow Chart da aplicação conjunta

3 Módulos

3.1 Classificador Naive-Bayes

Para treinar este módulo, utilizámos dois datasets: um contendo notícias falsas e o outro notícias verdadeiras, ambos com aproximadamente 7 mil notícias cada.

Inicialmente, removemos stop-words e todas as palavras com menos de 3 ocorrências em toda a data para reduzir o ruído nos dados. Com os dados processados, dividimos os datasets: 80% das notícias foram reservadas para treino e os 20% restantes para testes.

Durante o treino, calculámos as probabilidades necessárias utilizando o modelo Naive Bayes. Em seguida, aplicámos o classificador às notícias restantes para avaliar se os resultados obtidos correspondiam às expectativas.

3.2 Bloom Filters

Na sua preparação, carregamos o ficheiro CSV do Dataset de Fake News que obtivemos, capturando os links dos sites em questão. O Bloom Filter é, depois, iniciado com os parâmetros ideais e os links falsos são inseridos para treino. É, de seguida, feita uma verificação ao link fornecido e, caso não seja encontrado no Bloom Filter, a mensagem será, então, passada para o Classificador Naive Bayes e o Minhash. Caso contrário, o programa termina, classificando a notícia como falsa. Se o classificador determinar a notícia como falsa, ele insere no bloom filter a fonte passada anteriormente.

Optámos por incluir apenas fontes associadas a notícias falsas no filtro, pois, no contexto da nossa aplicação, é preferível que uma fonte verdadeira seja classificada como falsa do que o contrário.

3.3 Minhash

Como esta aplicação utiliza MinHash para otimizar o treino do classificador Naive Bayes e remover ruído, começámos por desenvolver as funções necessárias para gerar shingles (substrings de texto) para cada notícia e criar as assinaturas correspondentes.

Após implementar essas funções, aplicámo-las ao dataset de treino, gerando os shingles e as respetivas assinaturas para todas as notícias.

4 Testes

Nota: Adicionar as paths dos diretórios antes de correr os programas, ou através do "addpath('diretório')" no terminal, ou através do ficheiro "addpathAuto.m".

4.1 Teste do Classificador de Naive Bayes

Para verificar se o classificador está a classificar corretamente as notícias, realizámos testes utilizando um conjunto de notícias de teste. Estas notícias estão armazenadas num cell array, onde aproximadamente as primeiras 4500 linhas correspondem a notícias verdadeiras e as restantes a notícias falsas. Organizámos os dados desta forma para facilitar a verificação do desempenho do classificador.

As duas capturas de ecrã seguintes apresentam a classificação atribuída pelo classificador Naive Bayes às primeiras e às últimas notícias do conjunto de teste.

	1	2	3	4	5	6
1	Real	Real	Real	Real	Real	Real

Figure 2: Classificação das primeiras notícias

8801	8802	8803	8804	8805	8806
Fake	Fake	Fake	Fake	Fake	Fake

Figure 3: Classificação das últimas notícias

Como sabemos que cerca das 4500 primeiras notícias são verdadeiras e o resto falsas, facilitou a visualização da taxa de acerto executando o seguinte script:

```
% Gerar as classes verdadeiras (ground truth)
trueClasses = [repmat({'Real'}, 1, length(realTestNews)), ... % Classes reais
               repmat({'Fake'}, 1, length(fakeTestNews))]; % Classes falsas

numNews = length(testNews); % Number of news to be classified
classes = cell(1,numNews); % Initialize class

for newIdx = 1 : length(testNews) % For each new
    new = testNews(newIdx); % Extract current new

    % Classify the new
    class = classifyNew(new, uniqueWords, probWordReal, probWordFake, probFake, probReal);

    % Store the class
    classes{1,newIdx} = class;
end

% Comparar as classes preditas com as verdadeiras
correctPredictions = strcmp(classes, trueClasses); % Compara strings
numCorrect = sum(correctPredictions); % Total de classifica es corretas
accuracy = numCorrect / numNews; % Taxa de acerto
errorRate = 1 - accuracy; % Taxa de erro
```

Ao exibir os resultados, aparece o seguinte output:

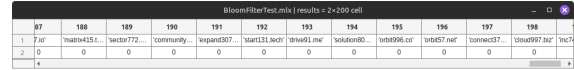
```
Número total de notícias testadas: 2993
Número de classificações corretas: 2849
Taxa de acerto: 95.19%
Taxa de erro: 4.81%
```

Figure 4: Assinaturas de algumas notícias teste

4.2 Teste do Bloom Filter

De maneira a realizar o teste de forma correta, dirija-se para a pasta própria do módulo do Bloom Filter e execute o programa de testes "BloomFilterTest.m". O teste consiste em duas funções sem parâmetros de entrada que devolvem um array cada (uma delas comentada de cada vez). Um dos arrays possui cerca de 100 elementos todos pertencentes ao Dataset usado para treinar o BloomFilter, e o outro possui cerca de 200 elementos não pertencentes, para calcular os falsos positivos.

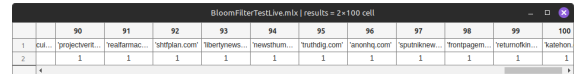
```
A notícia do site "drive91.me" não está registada como possível fake news!
A notícia do site "solution804.org" não está registada como possível fake news!
A notícia do site "orbit996.co" não está registada como possível fake news!
A notícia do site "orbit57.net" não está registada como possível fake news!
A notícia do site "connect373.tv" não está registada como possível fake news!
A notícia do site "cloud997.biz" não está registada como possível fake news!
A notícia do site "inc74.biz" não está registada como possível fake news!
A notícia do site "my412.tech" não está registada como possível fake news!
Logical Ones (1): 3
Logical Zeros (0): 197
False Positives: 1.5%
end of test
```



	87	188	189	190	191	192	193	194	195	196	197	198
1	1	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0

Figure 5: Teste de inclusão das últimas notícias do array das pertencentes.

```
A notícia do site "libertynews.com" pode ser fake news!
A notícia do site "newsthump.com" pode ser fake news!
A notícia do site "truthdig.com" pode ser fake news!
A notícia do site "anonhq.com" pode ser fake news!
A notícia do site "sputniknews.com" pode ser fake news!
A notícia do site "frontpagemag.com" pode ser fake news!
A notícia do site "returnofkings.com" pode ser fake news!
A notícia do site "katehon.com" pode ser fake news!
Logical Ones (1): 100
Logical Zeros (0): 0
end of test
```



	90	91	92	93	94	95	96	97	98	99	100
1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1

Figure 6: Teste de inclusão das últimas notícias do array das não pertencentes.

Em ambos os testes, após o cálculo dos valores de m, n e k ideais, obtivemos resultados bastante perto do esperado. A percentagem de falsos positivos, por exemplo, é de 1.5%, bastante perto do valor teórico esperado de 1%.

4.3 Teste do MinHash

Para testar o Minhash gerámos shingles e assinaturas para 200 notícias diferentes e verificámos valores válidos nas assinaturas (Na imagem seguinte podemos ver alguns dos valores gerados)

```
k = 5; % Size of shingles
hf = 300; % Number of hash functions
p = 1e9 + 7; % High prime number
R = randi(p, hf, 5); % Matrix of random number for
% each hash function

signatures = zeros(hf, 200); % Initialize signatures matrix

for i = 1:200 % Loop to iterate through test news

    % Get shingles of the current new
    shingles = generateShingles(testNews{i}, k);

    % Get current shingles's signatures
    signature = minhashSignatures(shingles, hf, p, R);

    % Store in the matrix
    signatures(:, i) = signature;
```

end

	1	2	3	4	5	6	7	8	9
5	826203	14635267	1687664	1138485	6949477	38399	4693913	445655	268196
6	1468985	201424	681950	607387	165008	607387	1530657	607387	12596
7	1161946	24208752	1506024	2194366	5357558	1161946	1372726	2194366	962110
8	412988	7322540	261573	491077	428185	23884	16025909	23884	256869
9	1148881	4319266	2208648	1687529	1646961	610466	6814376	53891	836430
10	1001962	2791818	485162	197599	2861823	238312	5602565	220763	185113
11	934279	7167337	1309817	1503249	2980407	1982522	2484591	2378079	319276
12	121539	2181579	880987	1707279	1775502	323562	93630	93630	243803
13	200868	1445654	806476	66019	1454682	66019	5587136	384632	124146
14	1300502	1052255	928481	5466680	18212996	4962560	10984198	571373	625932
15	1428021	4011090	278374	3811735	828855	724803	6083018	300098	243810
16	721051	1482524	1023353	494072	721051	224676	1799568	224676	423170
17	318041	6756794	318041	232577	70705	3016590	5074531	434926	816093

Figure 7: Assinaturas de algumas notícias teste

De seguida, testámos a similaridade da primeira notícia com todas as outras no banco de teste e obtivemos este resultado:

```

Top 5 Real News mais semelhantes:
New : trump focus program solely islam
Similaridade: 0.15
New : concerned experts add censorship
Similaridade: 0.14
New : trump likely face questions travel
Similaridade: 0.14
New : nobel peace prize winners criticized
Similaridade: 0.12
New : trump supreme court nominee goes
Similaridade: 0.12

```

Figure 8: Top 5 notícias verdadeiras semelhantes

```

Top 5 Fake News mais semelhantes:
New : another win court rules cities protest
Similaridade: 0.10
New : horrifying reality executions state
Similaridade: 0.09
New : trump claims hillary involved disappearance
Similaridade: 0.09
New : nra sponsor really loves trashing
Similaridade: 0.09
New : president obama donald trump treatment
Similaridade: 0.09

```

Figure 9: Top 5 notícias falsas semelhantes