Um Classificador de Seriedade de Notícias Baseado em Análise de Texto

Gustavo F. Camilo e Leonardo G. C. e Silva

17 de outubro de 2019

1 Introdução

A linguagem Perl proporciona diversas facilidades para manipulações de *strings*, o que justifica sua ampla adoção para processamento de texto. Aproveitando dessa qualidade, este relatório descreve a parte 2 do trabalho que consiste na implementação em Perl de um classificador de seriedade de notícias. Com esse objetivo, o programa classifica o assunto da notícia e calcula métricas ponderadas baseadas na quantidade de palavras mais utilizadas em cada assunto (celebridade, economia, política, ciência e esporte) para definir se a notícia é séria (retorno 0) ou não séria (retorno 1). Toda implementação e os arquivos auxiliares estão disponíveis no Github pelo endereço *https://github.com/JoltLeo/News_analyser*.

2 Implementação do Programa

O trabalho desenvolvido usou a linguagem Perl[1] para a implementação da análise de texto das notícias. Essa etapa do trabalho implementa cinco funções de análise de texto de notícias, para uma classificação do tópico da notícia e da seriedade da notícia. As cinco funções são descritas abaixo:

- **Contar o número de artifícios menos sérios utilizados na notícia**: recebe o arquivo texto a ser analisado como entrada. Essa função busca por palavras ou artifícios que aparecem constantemente em notícias menos sérias, e.g, palavras de baixo calão, *emoticons*, uso de primeira pessoa denotando opinião e uso de superlativos, constantemente usados em notícias menos sérias.
- **Verificar se o autor/site está na lista de autores/sites menos confiáveis**: recebe o arquivo texto a ser analisado. Essa função verifica se o nome do autor ou da fonte da notícia está presente na lista de autores ou de fontes menos confiáveis.
- **Buscar palavras-chave para a classificação do assunto da notícia**: recebe o arquivo texto com a notícia a ser analisada. Essa função busca por palavras-chave para classificar o tópico da notícia em esporte, economia, política, celebridades/fofoca, crime, ciência ou indefinido.
- **Classificador de seriedade da notícia**: recebe as métricas geradas pelas outras funções. Usando essas métricas com pesos de acordo com o tópico da notícia, a função atribui uma nota à notícia de 0 a 10.
- **Verificação se a notícia possui data e autor:** recebe o arquivo texto com a notícia a ser analisada. Muitos boatos e notícias não sérias omitem o nome do autor e a data em que foi escrita a notícia. Tendo isso em mente, essa função verifica se a notícia possui o autor na segunda linha e a data na terceira linha.

[1] Disponível em https://www.perl.org/

O trabalho foi separado em cinco módulos Perl, cada um implementando um das cinco funções descritas acima. Os cinco módulos feitos foram:

- check_artifacts.pm;
- check_blacklist.pm;
- check_news_format.pm;
- news_subject_classification.pm;
- classify_news.pm.

Além desses cinco módulos, o sistema implementa um menu para facilitar a interação do programa com o usuário nessa etapa do trabalho.

2.1 Estrutura dos Diretórios e Arquivos

Por decisão de projeto e organização, todos os códigos e arquivos estão no diretório *src/perl/*, e os módulos criados em Perl estão em *src/perl/modules*. Todos os módulos recebem um arquivo texto contendo uma notícia em português para ser analisada, e podem receber um arquivo texto contendo a lista negra de autores e fontes de notícias. A notícia deve conter o título, autor, data de publicação, fonte na primeira, segunda, terceira e quarta linha respectivamente. O conteúdo da notícia deve estar presente a partir da quinta linha. O arquivo da lista negra de autores/fontes apresenta um nome por linha. Como facilidade para execução do programa, o arquivo padrão da lista negra está disponível em *src/perl/blacklist.txt*.

2.2 O Módulo check_artifacts.pm

Este módulo é responsável pro checar o uso de artifícios não-sérios na notícia. Defini-se como artifícios não-sérios o uso de palavras de baixo-calão, uso de superlativos, *emoticons* e *emojis*, uso de primeira pessoa e a relação entre o número de palavras em caixa alta (*uppercase*) e palavras em caixa baixa (*lower case*). Esses artifícios foram escolhidos por estarem muito presentes em *fake news*. Cada um desses artifícios é implementado em uma sub-rotina, com o intuito de modularizar o programa.

A função **check_emoticons** procura por emoticons e emojis no texto da notícia. Para isso, a função implementa uma busca por expressão regular *regex*, para encontrar emoticons da forma: :), :D, :(, :p, :O, xD, XD. Além disso, a função procura por emojis. Como esses emojis não são escritos usando os caracteres convencionais do teclado, a busca se dá pelo Unicode do emoji. A Lista 1 no mostra a implementação da função check_emoticons. Os regex estão na linha 17 e 18 do programa. A linha 18 mostra a busca por emojis utilizando a faixa de Unicode que eles possuem. Esses artifícios são contados e guardados em um contador que é retornado ao final da função. Um peso maior foi dado aos emojis, uma vez que eles são muito usados em notícias falsas provenientes de redes sociais.

```
1  sub check_emoticons{
2      #Receives news file as argument
3      my $news_file_name = $_[0];
4      open (my $news, "<:encoding(UTF-8)", $news_file_name) or die "ERROR: Could not
       open file $news_file_name: $!\n";
5
6
7      my $temporary_emoji = 0;
8      my $temporary_emoticon = 0;
9      my $emoticon_counter = 0;
10     my $emoji_counter = 0;
11     my @emoticon_macthes;
12     my @emoji_matches;
13     my $counter = 0;
14
15     # Reads the news file and looks for emoticons and emojis
16     while (<$news>){
17         @emoticon_macthes = $_ =~ m/[:;][\)D\(\\pO][\s\n!]|[xX][D\\]/g;
```

```
        @emoji_matches = $_ =~ m/[\N{U+1F601}-\N{U+1F64F}]|[\N{U+2702}-\N{U+27B0}]|[\N{U+1F680}-\N{U+1F6C0}]/g; #Emojis are matched by their unicode

        $temporary_emoticon = scalar(@emoticon_macthes);
        $temporary_emoji = scalar(@emoji_matches);

        $emoticon_counter = $temporary_emoticon + $emoticon_counter;
        $emoji_counter = $temporary_emoji + $emoji_counter;
    }

    close $news or die "ERROR: Could not close file $news_file_name: $!\n";

    $counter = $emoticon_counter + 2*$emoji_counter;

    return ($counter);
}
```

Lista 1: Implementação da função check_emoticons.

A função **check_first_person** procura por pronomes de primeira pessoa no arquivo de notícias. Para isso, a função implementa uma busca por expressão regular buscando os pronomes de primeira pessoa. A Lista 2 apresenta a implementação da função.

```
#The check_first_person function looks for first person patterns in the news file
sub check_first_person{
    #Receives the news file as arguments
    my $news_file_name = $_[0];
    open (my $news, "<:encoding(UTF-8)", $news_file_name) or die "ERROR: Could not open file $news_file_name: $!\n";

    my $counter = 0;
    my @matches;
    my $temporary_counter = 0;

    while (<$news>){
        @matches = $_ =~ m/\b[Ee]u\b|\b[Mm]inha\b|\b[Mm]eu\b|\b[Nn]\N{U+00F3}s\b|\b[Aa]\sgente\b/g;
        $temporary_counter = scalar(@matches);

        $counter = $temporary_counter + $counter;
    }

    close $news or die "ERROR: Could not close file $news_file_name: $!\n";

    return $counter;
}
```

Lista 2: Implementação da função check_first_person.

A função **check_upper_to_lower_case_ratio** conta o número de ocorrências de caracteres em caixa alta e caixa baixa e retorna a relação entre eles. Esse critério foi adotado devido ao grande número de notícias falsas espalhadas utilizando muitas letras em caixa alta, principalmente em redes sociais, tentando gerar um tom alarmista. A Lista 3 mostra a implementação dessa função.

```
#The function chek_upper_to_lower_case_ratio calculates the upper to lower case ratio in the news file
sub check_upper_to_lower_case_ratio{
    #Receives the news file as argument
    my $news_file_name = $_[0];
    open (my $news, "<", $news_file_name) or die "ERROR: Could not open file $news_file_name: $!\n";

    my $line;
    my $char_counter = 0;
    my $uppercase_counter = 0;

    while ($line = <$news>){
        while ($line =~ m/\p{Uppercase}/g){
            $uppercase_counter ++;
        }
        $char_counter = $char_counter + length ($line);
```

```
    }

    my $upper_to_lower_ratio = $uppercase_counter/$char_counter;

    close $news or die "ERROR: Could not close file $news_file_name: $!\n";

    return $upper_to_lower_ratio;
}
```

Lista 3: Implementação da função check_upper_to_lower_case_ratio.

A função **check_superlative** procura por terminações que indicam o uso do superlativo em português. Para isso, a função implementa uma busca por expressão regular das terminações: -íssimo e -érrimo. A implementação da função pode ser encontrada na Lista 4.

```
#The check_superlative function counts the use of superlatives (via portuguese
    superlative ending matching) in the news file
#Inputs: news text file
sub check_superlative{
    my $news_file_name = $_[0];

    open (my $news, "<:encoding(UTF-8)", $news_file_name) or die "ERROR: Could not
    open file $news_file_name: $!\n";

    my @matches;
    my $temporary_counter = 0;
    my $counter = 0;

    while(<$news>){
        @matches = $_ =~ m/\N{U+00ED}ssimo\s|\N{U+00E9}rrimo\s/g;
        $temporary_counter = scalar(@matches);
        $counter = $temporary_counter + $counter;
    }

    close $news or die "ERROR: Could not close file $news_file_name: $!\n";

    return $counter;
}
```

Lista 4: Implementação da função check_superlative.

Por fim, a função **final_classifier** tem como objetivo juntar todas as outras funções do módulo check_artifacts para uma classificação final. Para isso, ela recebe como argumento o arquivo de notícias e o arquivo contendo as palavras de baixo calão. A função chama todas as outra funções do módulo e armazena os códigos de retorno em um vetor (*array*). A Lista 5 apresenta a implementação da função.

```
#The final_classifier metric receives the metrics calculated from the other
    functions and classifies the news as serious or not
#Inputs: news text file, curse_words text file
sub final_classifier{
    #Receives the metrics from the other functions to classify the seriousness
    metrics
     my @inputs = ($_[0], $_[1]);
     my @results = (check_emoticons($inputs[0]), check_first_person ($inputs[0]),
    check_upper_to_lower_case_ratio ($inputs[0]), check_curse_words (@inputs),
    check_superlative ($inputs[0]));

    return @results;
}
```

Lista 5: Implementação da função final_classifier.

2.3 O Módulo check_blacklist.pm

O módulo check_blacklist tem como objetivo verificar se o autor ou a fonte da notícia está marcada como não-confiável. Para isso, o módulo implementa uma função que verifica um arquivo contendo os autores e fontes não-confiáveis. Esse arquivo poderá ser modificado por um gerenciador

implementado em C++ na parte 3 do trabalho dessa disciplina. A função implementada assume que o autor está na segunda linha da notícia e fonte está na quarta linha da notícia, como descrito na parte 1 do trabalho. A Lista 6 apresenta a implementação da função check_blacklist.

```
1
2  package check_blacklist;
3
4  use strict;
5  use warnings;
6  use Exporter;
7  use check_news_format;
8
9  our @ISA= qw( Exporter );
10
11 # these CAN be exported.
12 our @EXPORT_OK = qw(check_author_and_source);
13
14 # these are exported by default.
15 our @EXPORT = qw(check_author_and_source);
16
17
18
19 #Function that verifies if the news author/source is on the blacklist of authors/
       source. Returns a metric for future evaluation of seriousness'
20 sub check_author_and_source{
21
22     #Receives news file and author blacklist file
23     my $news_file_name = $_[0];
24     my $blacklist_file_name = $_[1];
25     my $number_lines = $_[2];
26
27     my $returnCheck = check_news_format($news_file_name);
28     if ($returnCheck == -1) {
29         return -1;
30     }
31
32     open (my $news, "<", $news_file_name) or die "ERROR: Could not open file
       $news_file_name: $!\n";
33     open (my $blacklist, "<", $blacklist_file_name) or die "ERROR: Could not open
       file $blacklist_file_name: $!\n";
34
35     #Reads and discard the first and second line in order to get the author in the
       third line
36     my $line;
37     my $index;
38     for ($index = 0; $index < $number_lines; $index++){
39         $line = <$news>;
40     }
41
42     my $line2;
43     my $counter = 0;
44
45     #Compares each line from the blacklist with the authors line to verify if
       author is marked or not
46     while ($line2 = <$blacklist>) {
47         if ($line =~ /$line2/i ){
48             $counter = $counter + 1;
49         }
50     }
51
52
53     close $news or die "ERROR: Could not close file $news_file_name: $!\n";
54     close $blacklist or die "ERROR: Could not close file $blacklist_file_name: $!\n
       ";
55
56     return $counter;
57 }
```

Lista 6: Implementação da função check_blacklist.

2.4 O Módulo check_news_format.pm

O módulo check_news_format tem como objetivo verificar se o arquivo de notícias passado como argumento possui autor e data. Para isso, o arquivo verifica se a segunda e quarta linha do arquivo está vazia. Ainda, para a data, o arquivo verifica em qual formato a data está apresentada. A Lista 7 apresenta a implementação desse módulo.

```
1  package check_news_format;
2
3  use strict;
4  use warnings;
5  use Exporter;
6
7  our @ISA= qw( Exporter );
8
9  # these CAN be exported.
10 our @EXPORT_OK = qw(check_news_format);
11
12 # these are exported by default.
13 our @EXPORT = qw( check_news_format );
14
15
16 sub check_news_format{
17
18     my $news_file_name = $_[0];
19     open (my $news, "<", $news_file_name) or die "ERROR: Could not open file
       $news_file_name: $!\n";
20
21     #Verifies if there is at least one author in the news
22     my $line;
23     my $number_lines = 2;
24     my $index;
25     for ($index = 0; $index < $number_lines; $index++){
26         $line = <$news>;
27     }
28     chomp $line;
29     if ((not defined $line) || ($line eq "") || (length($line) == 0)){
30         close $news or die "ERROR: Could not close file $news_file_name: $!\n";
31         return -1;
32     }
33
34     #Matches date format in the third line of the news
35     my $counter = 0;
36     $line = <$news>;
37     if ($line =~ m/\d\d\/\d\d\/\d\d|\d\/\d\d\/\d\d|\d\/\d\/\d\d|\d\d\/\d\d\/\d\d\d\
       d|\d\/\d\d\/\d\d\d\d|\d\/\d\/\d\d\d\d/){
38         $counter ++;
39     }
40     if ($line =~ m/\d\d\-\d\d\-\d\d|\d\-\d\d\-\d\d|\d\-\d\-\d\d|\d\d\-\d\d\-\d\d\d\
       d|\d\-\d\d\-\d\d\d\d|\d\-\d\-\d\d\d\d/){
41         $counter ++;
42     }
43
44     if ($counter == 0){
45         close $news or die "ERROR: Could not close file $news_file_name: $!\n";
46         return -1;
47     }
48
49
50     #Verifies if there is a source in the news
51     $line = <$news>;
52     chomp $line;
53
54     if ((not defined $line) || ($line eq "") || (length($line) == 0)){
55         close $news or die "ERROR: Could not close file $news_file_name: $!\n";
56         return -1;
57     }
58
59     close $news or die "ERROR: Could not close file $news_file_name: $!\n";
60     return 0;
```

```
61 }
```

Lista 7: Implementação da função check_news_format.

2.5 O Módulo news_subject_classification.pm

O módulo news_subject_classification tem como objetivo classificar o assunto da notícia recebida como argumento. Devido ao tamanho do código, sua implementação está disponível apenas no Github[2]. Para determinar o tópico, o sistema analisa o texto da notícia buscando palavras-chave que o ajudem na classificação do assunto. As palavras-chave usadas foram obtidas através da análise de diversas notícias de diferentes temas e fontes. O trabalho utilizou o Apache Spark[3] [1], um *framework* de computação distribuída para processamento de grande volume de dados (*big data*), para contar as palavras da notícia. O código de contagem de palavras no Spark foi implementado em Python[4] e está descrito na Lista 8 do Apêndice A. As palavras relevantes com maiores ocorrências foram selecionadas para serem utilizadas na busca por expressão regular. A função **classify_subject** chama 5 sub-rotinas para avaliar a notícia mediante ao seu possível assunto. As sub-rotinas seguem a seguinte lógica:

- **check_celebrity_subject** avalia a existência das palavras "gente", "você(s)", "novela(s)", "famoso(s) ou famosa(s)", "expectativa", "filho(s) ou filha(s)", "!", "casamento" e "festa" para avaliar a chance da notícia ser sobre celebridades;
- **check_economy_news** avalia a existência das palavras "reforma(s)", "banco(s)", "$ ou dolar(es)", "mercado(s)", "dinheiro(s)", "empresa(s)", "economia(s)", "tributo(s) ou tributário(s) ou tributária(s)" e "% ou juros" para avaliar a chance da notícia ser sobre economia;
- **check_politics_news** avalia a existência das palavras "presidente(s)", "ministro(s) ou ministério(s)", "senhor", "governo(s) ou governador(es) ou governadora(s)", "supremo", "partido(s) ou partidário(s) ou partidária(s)", "federal(is)", "caso(s)" e "senado ou senador(es) ou senadora(s)" para avaliar a chance da notícia ser sobre política;
- **check_science_news** avalia a existência das palavras "tratamento(s) ou tratado(s)", "pesquisa(s) ou pesquisador(es) ou pesquisadora(s)", "universidade", "conjugações do verbo descobrir ou descoberto(s) ou descoberta(s)", "nobel", "prêmio(s)", "tecnologia(s)", "doença(s)" e "novo(s) ou nova(s)" para avaliar a chance da notícia ser sobre ciência;
- **check_sports_news** avalia a existência das palavras "mundo(s) ou mundial(is)", "primeiro(s) ou primeira(s)", "time(s) ou seleção ou seleções", "final(is)", "carreira(s)", "atleta(s)", "conjugações do verbo jogar ou jogador(es) ou jogadora(s)", "conjugações do verbo treinar ou treinador(es) ou treinadora(s) ou treinamento(s)" e "conjugações dos verbos vencer e ganhar" para avaliar a chance da notícia ser sobre esporte;

Para cada uma das possibilidades, o módulo implementa uma média ponderada para atribuir uma nota. Os pesos são de 1 à 9 e são aplicados no contador de cada uma das palavras selecionadas para busca de acordo com a avaliação feita no Spark [1]. Portanto, para cada assunto, as palavras mais reincidentes recebem os maiores pesos. Por fim, a função **classify_subject** retorna uma string com o nome do assunto que obteve a maior das notas atribuídas pelas sub-rotinas.

2.6 O Módulo classify_news.pm

Este módulo implementa a função classify_news que tem como objetivo classificar a seriedade da notícia. A função recebe como entrada o arquivo texto contendo a notícia e pode receber os arquivos texto da lista negra de autores/fontes. Devido ao tamanho do código, sua implementação está disponível apenas no Github[5]. A sub-rotina classify_news chama todas as outras funções para

[2]Disponível em https://github.com/JoltLeo/News_analyser/blob/master/src/perl/modules/news_subject_classification.pm
[3]Disponível em http://spark.apache.org/
[4]Disponível em https://www.python.org/
[5]Disponível em https://github.com/JoltLeo/News_analyser/blob/master/src/perl/modules/classify_news.pm

obter o assunto da notícia e suas métricas para avaliação. As métricas são compostas pelo número de *emoticons*, número de termos de 1ª pessoa, a relação entre o número de ocorrências de caracteres em caixa alta e caixa baixa, número de palavras de baixo calão e número de superlativos. Se o autor ou a fonte estiver presente na lista negra, a notícia é imediatamente considerada não séria. Caso contrário, a classificação é feita utilizando estatísticas feitas previamente no Spark. Com isso, a notícia é classificada como séria se suas métricas condizem com as estatísticas do assunto da notícia. Ao final da execução da função classify_news, O assunto da notícia, sua classificação e todas as avaliações e métricas são impressos na tela.

2.7 O Menu de Uso

O sistema implementa um menu para melhor interação do usuário com o programa. Quando o programa é inicializado, o menu é impresso na tela mostrando as opções ao usuário. As cinco funções apresentadas na seção anterior e a opção de sair do programa são apresentadas. Selecionada uma opção, o usuário deve passar o arquivo contendo a notícia a ser analisada. O programa também dá ao usuário a opção de usar uma lista própria com fontes não-confiáveis. Caso o usuário não possua uma lista própria, o programa utilizará a lista padrão. Devido ao tamanho do código, sua implementação está disponível apenas no Github[6].

3 Casos de Uso

Este trabalho aceita somente notícias em língua portuguesa e em arquivos texto. A interação do usuário com o sistema é dado através do menu. A Figura 1 mostra como é feita a interface entre o usuário e o programa. Para a execução do programa, o usuário deve executar o comando `perl menu.pl` dentro do diretório que contém esse arquivo.

```
--------------------------------------------------------------------------------
                              Menu
--------------------------------------------------------------------------------
|Type 0 to quit the program                                                    |
|Type 1 to check if the author/source is on the blacklist                      |
|Type 2 to classify the subject of the news                                    |
|Type 3 to verify if the news has author and date                              |
|Type 4 to count artifacts less serious on the news                            |
|Type 5 to classify if the news is serious or NOT serious                      |
--------------------------------------------------------------------------------
NOTE: this program do not interpret English news, you MUST enter a news text file written in PORTUGUESE when asked!

Type the desired option and press enter: 
```

Figura 1: Menu do sistema implementado.

O usuário deve escolher a opção desejada. Caso escolha sair do programa, uma mensagem de despedida é impressa na tela como a da Figura 2a. Caso selecione uma opção não apresentada na tabela, uma mensagem de erro como a da Figura 2b é impressa na tela.

```
Type the desired option and press enter: 0
--------------------------------------------------------------------------------
                              Goodbye!
--------------------------------------------------------------------------------
```

(a) Mensagem mostrando uma mensagem de adeus.

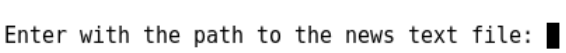
```
Type the desired option and press enter: g
--------------------------------------------------------------------------------
  This is not a valid option. Type an option from the menu! ¯\_(ツ)_/¯
--------------------------------------------------------------------------------
```

(b) Mensagem de erro mostrando que a tecla apertada não é uma opção do menu.

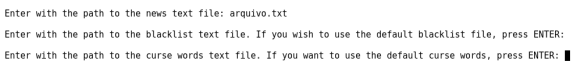
Caso o usuário selecione uma opção do menu, como a de classificar o assunto da notícia, o programa solicita os arquivos necessários para a execução do programa, como o arquivo de notícias, o arquivo contendo a lista de autores e fontes marcadas como não-confiáveis. Caso o usuário não possua os dois últimos arquivos, o programa é executado utilizando os arquivos padrões. Essa característica do sistema permite uma melhor personalização, melhorando a experiência do usuário e adequando o sistema às necessidades de quem usa. A Figura 3a mostra o programa pedindo a

[6]Disponível em https://github.com/JoltLeo/News_analyser/blob/master/src/perl/menu.pl

entrada do arquivo quando o usuário seleciona a opção de classificar o assunto da notícia. O programa pede somente o arquivo de notícias, uma vez que para classificar o assunto da notícia só é necessário este arquivo. A Figura 3b mostra o programa pedindo todos os arquivos, quando a opção cinco do menu é selecionada, buscando uma classificação geral.



(a) Programa pedindo que o usuário entre com o arquivo contendo uma notícia.



(b) Programa pedindo que o usuário entre com os arquivos necessários.

Para testar as funcionalidades, foi utilizado a notícia sobre política do site globo.com[7] com maior destaque no momento do teste[8]. A Figura 4 mostra a notícia em arquivo texto no formato especificado anteriormente.

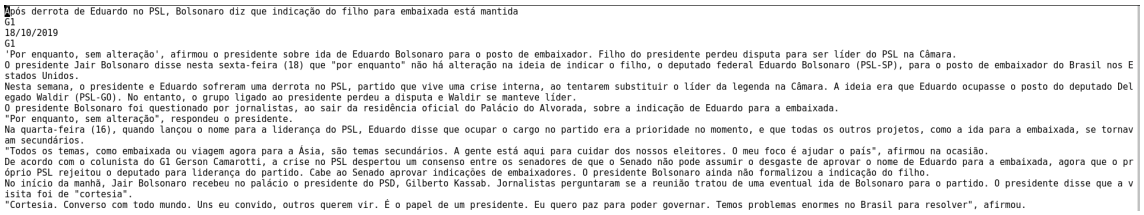

Após derrota de Eduardo no PSL, Bolsonaro diz que indicação do filho para embaixada está mantida
G1
18/10/2019
G1
'Por enquanto, sem alteração', afirmou o presidente sobre ida de Eduardo Bolsonaro para o posto de embaixador. Filho do presidente perdeu disputa para ser líder do PSL na Câmara.
O presidente Jair Bolsonaro disse nesta sexta-feira (18) que "por enquanto" não há alteração na ideia de indicar o filho, o deputado federal Eduardo Bolsonaro (PSL-SP), para o posto de embaixador do Brasil nos Estados Unidos.
Nesta semana, o presidente e Eduardo sofreram uma derrota no PSL, partido que vive uma crise interna, ao tentarem substituir o líder da legenda na Câmara. A ideia era que Eduardo ocupasse o posto do deputado Delegado Waldir (PSL-GO). No entanto, o grupo ligado ao presidente perdeu a disputa e Waldir se manteve líder.
O presidente Bolsonaro foi questionado por jornalistas, ao sair da residência oficial do Palácio do Alvorada, sobre a indicação de Eduardo para a embaixada.
"Por enquanto, sem alteração", respondeu o presidente.
Na quarta-feira (16), quando lançou o nome para a liderança do PSL, Eduardo disse que ocupar o cargo no partido era a prioridade no momento, e que todas os outros projetos, como a ida para a embaixada, se tornavam secundários.
"Todos os temas, como embaixada ou viagem agora para a Ásia, são temas secundários. A gente está aqui para cuidar dos nossos eleitores. O meu foco é ajudar o país", afirmou na ocasião.
De acordo com o colunista do G1 Gerson Camarotti, a crise no PSL despertou um consenso entre os senadores de que o Senado não pode assumir o desgaste de aprovar o nome de Eduardo para a embaixada, agora que o próprio PSL rejeitou o deputado para liderança do partido. Cabe ao Senado aprovar indicações de embaixadores. O presidente Bolsonaro ainda não formalizou a indicação do filho.
No início da manhã, Jair Bolsonaro recebeu no palácio o presidente do PSD, Gilberto Kassab. Jornalistas perguntaram se a reunião tratou de uma eventual ida de Bolsonaro para o partido. O presidente disse que a visita foi de "cortesia".
"Cortesia. Converso com todo mundo. Uns eu convido, outros querem vir. É o papel de um presidente. Eu quero paz para poder governar. Temos problemas enormes no Brasil para resolver", afirmou.

Figura 4: Notícia utilizada para o teste de classificação.

Executando a opção cinco para uma classificação geral da notícia, é possível ver na Figura 5 que o programa classifica corretamento o assunto da notícia em política, imprime todas as métricas utilizadas na classificação de seriedade.

```
Using default text file for blacklist

Using default text file for curse words

The arquivo.txt subject is politics.
In arquivo.txt, there are:

The author is NOT on the blacklist;
The sourcer is NOT on the blacklist;
0 emoticons;
4 first person;
0.0588487972508591 upper to lower case ratio;
0 curse words;
0 superlatives


This politics news arquivo.txt is serious.
```

Figura 5: Apresentação da classificação final da notícia.

Um segundo teste foi executado com uma notícia do site Sensacionalista[9], site conhecido por postar notícias falsas e satíricas do dia-a-dia. A notícia escolhida foi a que estava como destaque no momento do teste[10]. No site de onde a notícia foi selecionada, não há informações sobre o autor ou a data, logo as linhas correspondentes a esse campo foram deixadas em branco. O programa detecta que faltam informações para classificar a notícia e mostra uma mensagem de erro avisando ao usuário que a notícia não esta no formato correto. A Figura 6 mostra a mensagem de erro no terminal.

[7] https://www.globo.com/

[8] https://g1.globo.com/politica/noticia/2019/10/18/apos-derrota-de-eduardo-no-psl-bolsonaro-diz-que-indicacao-do-filho-para-embaixada-esta-mantida.ghtml

[9] https://www.sensacionalista.com.br/

[10] https://www.sensacionalista.com.br/2019/10/18/bolsonaro-e-chamado-de-vagabundo-e-rebate-dizendo-que-aprovou-2-projetos-em-27-anos-de-congresso/

```
News text file noticia.txt in wrong format! Check documentation at https://github.com/JoltLeo/News_analyser
```

Figura 6: Apresentação da classificação final da notícia.

Para contornar esse problema, o arquivo de notícias foi modificado para que no lugar do autor, aparecesse a fonte e a data adicionada foi a do dia do teste da notícia. Dessa maneira, a fonte foi repetida na segunda e quarta linha. O programa classifica o assunto da norícia como política e verifica que o assunto não é sério. A Figura 7 mostra a execução desse cenário.

```
The noticia.txt subject is politics.
In noticia.txt, there are:

The author is on the blacklist;
The source is on the blacklist;
0 emoticons;
1 first person;
0.0452781371280724 upper to lower case ratio;
7 curse words;
0 superlatives


The author is on the blacklist, so the noticia.txt is not serious.
```

Figura 7: Execução do cenário descrito.

4 Conclusão

A proposta do trabalho é implementar um classificador de seriedade de notícias baseado em análise de texto. Este trabalho apresenta o módulo responsável pela análise do texto da notícia, sendo implementando em Perl através de buscas por expressões regulares. A classificação final da notícia é feita através de um conjunto de métricas calculadas pelo programa. Além disso, o sistema classifica o assunto de notícias em cinco categorias, através de buscas por palavras-chave no texto das notícias. A escolha das palavras foi feita através do um programa, com auxílio do *software* Apache Spark, que conta as palavras mais usadas em notícias de diversas categorias. O sistema foi testado com notícias de diversas áreas diferentes e apresentou bons resultados, classificando o assunto e a seriedade dos arquivos corretamente.

Referências

[1] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster Computing with Working Sets," in *Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing*, HotCloud'10, (Berkeley, CA, USA), pp. 10–10, USENIX Association, 2010.

A Implementação da Contagem de Palavras no Spark

```
#Execute code with "spark-submit --master spark://master:7077 <path to this python code>"

#Loading libs and Spark configurations
from pyspark import SparkContext, SparkConf
conf = SparkConf().setAppName("word_count")
sc = SparkContext(conf=conf)

#Input file to spark read
text_file = sc.textFile("hdfs://master:9000/user/app/politics_news.txt")
counts = text_file.flatMap(lambda line: line.split(" ")) \
             .map(lambda word: (word, 1)) \
             .reduceByKey(lambda a, b: a + b, 1) \
             .map(lambda (a, b): (b, a)) \
             .sortByKey(0, 1) \
             .map(lambda (a, b): (b, a))


#Output directory to save results
counts.saveAsTextFile("hdfs://master:9000/user/app/politics_count.txt")
```

Lista 8: Implementação da função de contagem de palavras no Spark.