

Um Classificador de Seriedade de Notícias Baseado em Análise de Texto

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

25 de Novembro de 2019

1 Introdução

O paradigma de programação orientada a objetos é inserida na linguagem C, sendo esta versão chamada de C++. A possibilidade do uso e criação de Classes elevam o grau de modularidade dos códigos e facilita o entendimento do programa como um todo. Dessa forma, aproveitando dessas qualidades, este relatório descreve a **parte 3** do trabalho que consiste na implementação em C++ de uma interface com um classificador de seriedade de notícias em Perl. Além disso, também propõem-se 4 funcionalidades extras escritas em C++ para o gerenciamento dos arquivos e criação de uma interface amigável com o usuário. 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 C++ para a implementação de um gerenciador de um sistema que classifica a seriedade das notícias. Essa etapa do trabalho implementa cinco funções de gerenciamento de notícias e do sistema para classificar o tópico e a seriedade de uma notícia. Além disso, essa etapa integra o gerenciador feito em linguagem C++ com o analisador de texto implementado em Perl da etapa anterior e implementa uma interface amigável com o usuário. As cinco funções são descritas abaixo:

- **Exibição de uma notícia:** recebe como entrada uma *string* com o nome do arquivo de texto em da notícia e não possui retorno. Caso a notícia seja encontrada, seu conteúdo é impresso na tela, se não uma mensagem de erro é impressa na tela;
- **Listar os títulos das notícias classificadas como menos sérias:** não recebe argumentos do usuário e não possui retorno. A função imprime os títulos de todas as notícias classificadas como menos séria na tela;
- **Classificar a seriedade de uma ou todas as notícias:** pode receber ou não argumentos do usuário. Essa função chama o analisador de texto implementado em Perl para classificar a notícia. A função possui argumento opcional do nome do arquivo texto com a notícia a ser classificada. Caso não receba esse argumento, o modulo Perl classifica todas as notícias do diretório padrão de notícias. A função imprime na tela o número de notícias sérias e não sérias;
- **Adicionar ou remover autor/site da lista de autores/sites menos confiáveis:** recebe como argumento a operação, i.e, adicionar ou remover, além de receber o nome do autor ou endereço do site a ser adicionado ou removido e imprime na tela uma mensagem de sucesso ou erro;
- **Listar o número total de notícias e a relação das classificações:** não recebe argumentos do usuário e não possui retorno. Essa função imprime na tela uma tabela contendo o número total de notícias classificadas, o número de notícias classificadas como sérias e o número de notícias classificadas como menos sérias.

O trabalho foi separado em oito arquivos escritos em C++, cinco implementando cada uma das cinco funções, um menu para interação com o usuário, um programa principal que implementa a proposta e outro que implementa a classe de erros. Os oito são listados e resumidos abaixo:

- **show_news.cpp**: Implementa a função que exibe uma notícia ao usuário dado o caminho do arquivo.
- **classification_statistics.cpp**: Implementação da função `classification_statistics` que recebe o nome do arquivo com as classificações das notícias para mostrar ao usuário informações sobre a classificação.
- **change_blacklist.cpp**: Implementação de uma função que altera a lista contendo o nome de fontes menos confiáveis. Essa função permite que usuários possam modificar a lista, gerando uma experiência mais personalizada as suas necessidades.
- **error_class.cpp**: Implementação da classe *Error_class* que identifica se o retorno de uma função é de fato um erro. Caso positivo, imprime na tela a mensagem de erro referente ao código de retorno da função;
- **list_less_serious.cpp**: Implementação da função `list_less_serious` que lista as notícias classificadas como menos sérias por título.
- **perl_wrapper.cpp**: Implementação da classe da interface, *Perl_wrapper*, entre o C++ e o Perl. Essa classe faz o gerenciamento dos parâmetros passados e os retornos da interface do Perl;
- **menu.cpp**: Implementação da classe *Menu* que gerencia a impressão do menu principal do programa, a entrada de dados pelo usuário, as chamadas dos outros módulos que realizam as funcionalidades propostas para a parte do programa em C++ e a análise e tratamento dos erros;
- **main.cpp**: Programa principal que utiliza um objeto da classe *Menu* para executar métodos que realizam as funcionalidades propostas de acordo com a opção do menu escolhida pelo usuário.

O programa principal deve ser criado pelo *src/C++/Makefile* ao executar o comando `'make install'` no diretório *src/C++/*, criando o executável *news_analyser*. Para desinstalar o program, execute o comando *make clean* nesse mesmo diretório.

2.1 Estrutura dos Diretórios e Arquivos

Por decisão de projeto e organização, todos os códigos em C++ e arquivos de texto estão no diretório *src/C++/*, e os códigos usados para os testes estão em *src/C++/test.codes*. O arquivo *constantes.h* contem a declaração de todas as constantes utilizadas no programa, enquanto o arquivo *error_messages.h* mapeia todos os códigos de error do C++ e Perl e suas respectivas mensagens de retorno. O programa principal está implementado no arquivo *main.cpp* e a interface com o classificador de notícias em Perl é feita pela classe *Perl_wrapper*.

2.2 Implementação das Classes em C++

A implementação da interface contou com a criação de 3 classes escritas em C++, denominadas *Error_class*, *Perl_wrapper* e *Menu*.

2.2.1 Classe *Error_class*

A classe *Error_class* tem a finalidade de analisar os códigos de retorno das funções, identificar a respectiva mensagem de erro e realizar sua impressão na tela. Para a visualização das mensagens de erro, utilizou-se uma sobrecarga no operador *jj* da classe *ostream*. Os atributos e métodos da *Error_class* são apresentados no diagrama UML da Figura 1.

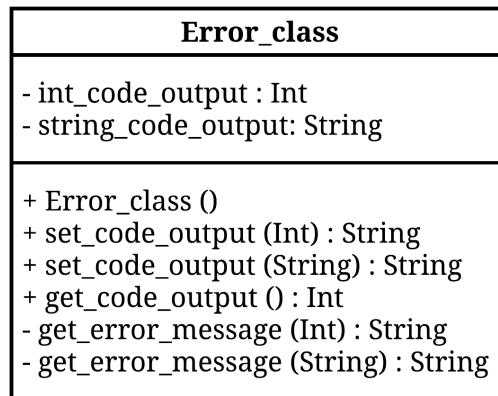


Figura 1: Diagrama UML da classe **Error_class**.

Vale ressaltar que a sobrecarga dos métodos *get_code_output* e *get_error_message* são realizadas para possibilitar o tratamento tanto de códigos de retorno do tipo *Int* quando do tipo *String*.

2.2.2 Classe *Perl_wrapper*

A classe *Perl_wrapper* possibilita a criação da interface com o classificador de notícias em Perl. Essa funcionalidade é realizada pelo método *classify_news (String)* que recebe como argumento uma *string* contendo o arquivo da notícia para análise. A interface acontece ao utilizar o modulo *classify_news.pm* escrito em Perl que utiliza os demais módulos apresentados na Parte 2 deste trabalho e realiza todos os procedimentos necessários para classificação da notícia. O método retorna um valor *Int*, sendo o retorno 0 uma notícia séria, 1 uma notícia NÃO seria e retornos negativos significam erro. Os atributos e demais métodos da *Perl_wrapper* são apresentados no diagrama UML da Figura 2.

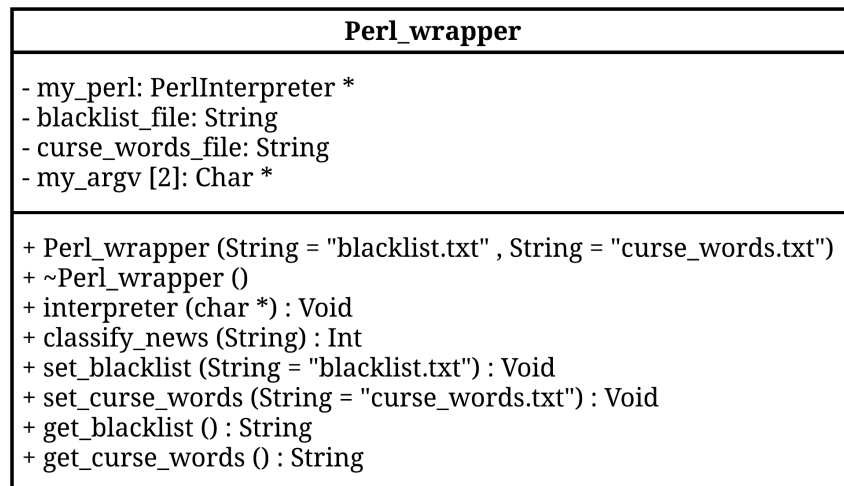


Figura 2: Diagrama UML da classe **Perl_wrapper**.

2.2.3 Classe *Menu*

A classe *Menu* busca simplificar o entendimento do programa e modular a exibição e execução das funcionalidades da interface C++. Essa classe realiza a impressão do menu principal na tela, gerencia todas as entradas do usuário ao programa, chama as funções para atender uma determinada

funcionalidade das 5 propostas. As opções do menu principal e sua execução serão demonstrados na Seção 3. Um detalhe importante dessa classe encontra-se no método *menu_classify_news ()* que cria um objeto *static* da classe *Perl_wrapper*. Essa abordagem foi necessária para evitar conflitos com o interpretador do Perl, visto que pode existir somente 1. Os atributos e demais métodos da *Menu* são apresentados no diagrama UML da Figura 3.

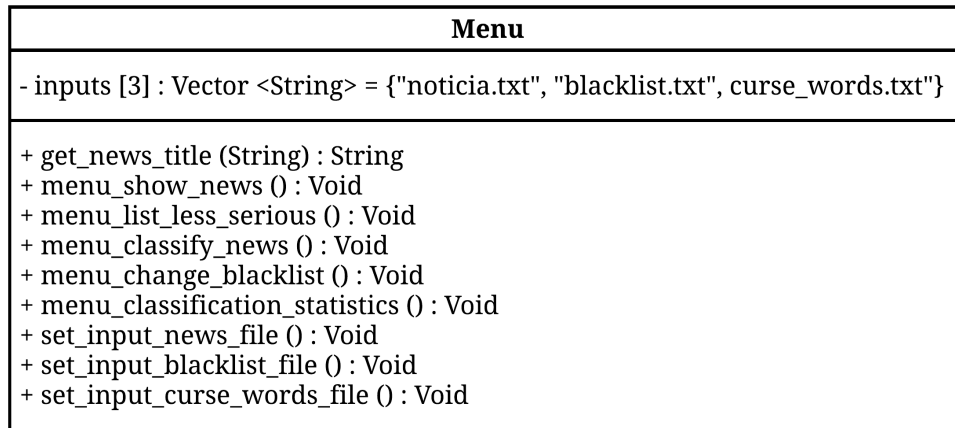


Figura 3: Diagrama UML da classe **Perl_wrapper**.

2.3 O Arquivo show_news.cpp

Esse arquivo implementa uma função que exibe a notícia ao usuário. Para isso, o programa recebe o caminho até o arquivo contendo a notícia e a exibe na tela. A Lista 1 apresenta o código dessa função.

```

1 #include <iostream>
2 #include <string>
3 #include <fstream>
4 #include "constants.h"
5 #include "show_news.h"
6
7 using namespace std;
8
9 int show_news (string news_filename){
10
11     ifstream file ;
12     string line ;
13
14     if (news_filename.length() == 0){
15         return BLANK_NEWS_FILENAME;
16     }
17
18     file.open(news_filename);
19     if (file.is_open()){
20         while (getline (file , line)) {
21             cout << line << endl;
22         }
23     } else {
24         //cout << "Could not open file " << news_filename << endl;
25         return NEWS_FILE_ERROR;
26     }
27     return SUCCESS;
28 }
29

```

Lista 1: Implementação da função show_news.

2.4 O Arquivo classification_statistics.cpp

Este arquivo é responsável por entregar ao usuários as estatísticas de notícias classificadas. Para isso, o programa lê um arquivo texto que guarda estatísticas sobre as notícias classificadas. A saída da função é uma tabela mostrando ao usuário o número total de notícias classificadas, o número de notícias classificadas como séria e o número de notícias classificadas como menos séria. A Lista 2 apresenta o código que implementa essas funcionalidades.

```
1 #include <string>
2 #include <iostream>
3 #include <fstream>
4 #include <iomanip>
5 #include "constants.h"
6 #include "classification_statistics.h"
7
8 using namespace std;
9
10 int classification_statistics(string filename){
11     unsigned counter_serious = 0;
12     unsigned counter_less_serious = 0;
13     unsigned ratio = 0;
14
15     size_t serious_position;
16     size_t less_serious_position;
17
18     string line;
19     ifstream classification_file;
20
21     classification_file.open(filename);
22     if (!classification_file.is_open()){
23         return CLASSIFICATION_FILE_ERROR;
24     }
25
26     while (getline(classification_file, line)){
27         serious_position = line.find(SERIOUS);
28         less_serious_position = line.find(LESS_SERIOUS);
29
30         if ((!serious_position) && (!less_serious_position)){
31             classification_file.close();
32             return CORRUPTED_FILE;
33         }
34         if (serious_position != string::npos){
35             counter_serious++;
36         } else {
37             counter_less_serious++;
38         }
39     }
40
41     cout << "_____" <<
42     endl;
43     cout << " |   NUMBER OF NEWS   |   SERIOUS NEWS   |   LESS SERIOUS NEWS |" <<
44     endl;
45     cout << "_____" <<
46     endl;
47     cout << setw(13) << counter_less_serious + counter_serious << setw(18) <<
48     counter_serious << setw(20) << counter_less_serious << endl;
49     cout << "_____" <<
50     endl;
51 }
```

Lista 2: Implementação da função classification_statistics.

3 Casos de Uso

O usuário interage com o sistema através de um menu que fornece as opções de sair do programa além das funcionalidades apresentadas anteriormente. A Figura ?? apresenta uma imagem do menu que faz a interface entre o usuário e o programa.

```

-----
Menu
-----
|Type 0 to quit the program
|Type 1 to show a news
|Type 2 to show the title of news classified as less serious
|Type 3 to classify the one or more news in serious or NOT serious
|Type 4 to add/remove an author from the blacklist
|Type 5 to show a relation of all classified news
-----
NOTE: this program do not interpret English news, you MUST enter a news text file written in PORTUGUESE when asked!

```

Figura 4: Menu do sistema implementado.

Ao selecionar uma opção, o programa pede ao usuário informações necessárias para executar a funcionalidade. Caso o usuário prefira, o programa é executado com as opções padrão. A Figura 5 mostra como o programa pede ao usuário as entradas necessárias quando a opção 3 é selecionada. Essa opção comunica com o analisador em Perl da etapa anterior do trabalho para classificar as notícias como sérias ou não sérias.

Option 3 choosen

```

Please, enter the path to the news file or just press enter to use default path:
Please, enter the path to the blacklist file or just press enter to use default path:
Please, enter the path to the curse words file or just press ENTER to use default path:

```

Figura 5: Programa pedindo que o usuário forneça as entradas necessárias.

Como apresentado na etapa anterior do trabalho, essa opção apresenta as estatísticas coletadas pelo programa que foram usadas para classificar a notícia. A Figura 6 apresenta a notícia usada para o teste que foi impressa na tela utilizando a opção 1 do menu.

```

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 Bolso
naro (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 mant
eve líder.
O presidente Bolsonaro foi questionado por jornalistas, ao sair da residência oficial do Palácio do Alvorada, sobre a indicação de Eduardo para a embaixad
a.
"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 o
s 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 ajuda
r 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 a
provar 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 emb
aixadores. 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 id
a 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 enorm
es no Brasil para resolver", afirmou.

```

Figura 6: Notícia usada para o teste.

Os resultados obtidos com a notícia da Figura 6 são mostrados na Figura 7.

```
The noticia.txt subject is politics.
In noticia.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 noticia.txt is serious.
```

Figura 7: Resultado da notícia classificada.

Utilizando a opção 4 para modificar a lista com as fontes menos confiáveis, o programa fornece ao usuário as opções de remover ou adicionar uma fonte à lista, além da opção de sair do programa. Selecionada a opção de adicionar ou remover um nome da lista, o programa pede ao usuário o nome a ser removido ou adicionado. Adicionando a fonte da notícia da Figura 6 e executando o programa novamente, a saída muda. As Figuras 8a e 8b apresentam as opções quando a opção 4 é selecionada e a saída do programa quando ele é executado novamente com as novas mudanças, respectivamente.

```
Option 4 choosen

Please, enter the path to the blacklist file or just press enter to use default path:
Type "a" to add an/a author/source to the blacklist file
Type "r" to remove an/a author/source to the blacklist file
Type "e" to exit
a
Type the name you want to add to the blacklist: GI

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;
4 first person;
0.0588487972508591 upper to lower case ratio;
0 curse words;
0 superlatives

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

(a) Programa pedindo que o usuário entre com o nome a ser acrescentado à lista.

(b) Saída do programa mostrando a notícia como não séria, uma vez que a fonte foi acrescentada à lista de menos confiáveis.

As Figuras 9a e 9b apresentam a saída da opção 5 antes e depois da inclusão do autor à lista de menos confiáveis.

```
Option 4 choosen

Please, enter the path to the blacklist file or just press enter to use default path:
Type "a" to add an/a author/source to the blacklist file
Type "r" to remove an/a author/source to the blacklist file
Type "e" to exit
a
Type the name you want to add to the blacklist: GI

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;
4 first person;
0.0588487972508591 upper to lower case ratio;
0 curse words;
0 superlatives

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

(a) Estatísticas de classificação mostram uma notícia classificada como séria antes da inclusão do autor à lista de menos confiáveis.

(b) Estatísticas de classificação mostram uma notícia classificada como não séria após a inclusão do autor à lista de menos confiáveis.

Utilizando a opção 2, o programa exibe as notícias classificadas como menos sérias na tela pelo título. A Figura 10 mostra a saída do programa após a inclusão do autor da notícia à lista de fontes menos confiáveis.

```
Option 2 choosen
```

```
-----  
|               NOT SO SERIOUS NEWS BY TITLE               |  
-----  
"Após derrota de Eduardo no PSL, Bolsonaro diz que indicação do filho para embaixada está mantida"  
-----
```

Figura 10: Notícias classificadas como menos sérias por títulos.

4 Conclusão

A proposta do trabalho é implementar um classificador de seriedade de notícias baseado em análise de texto. Este trabalho apresenta sua parte 3 que consiste na implementação da interface C++ com o classificador de notícia escrito em Perl. A classificação final da notícia é feita através de um conjunto de métricas calculadas pelo programa em Perl. A eficácia da interface foi demonstrada e obteve sucesso na comunicação com o módulo Perl através do uso de um estrutura de dados do tipo pilha. Além da classificação correta do assunto e da seriedade da notícia previamente comprovado na parte 2 deste trabalho e da eficácia da interface do C++ com Perl, uma interface amigável e simples foi desenvolvida, assim facilitando o uso e entendimento do sistema

A Implementação da Contagem de Palavras no Spark

```
1 #Execute code with "spark-submit --master spark://master:7077 <path to this python
   code>"
2
3 #Loading libs and Spark configurations
4 from pyspark import SparkContext, SparkConf
5 conf = SparkConf().setAppName("word_count")
6 sc = SparkContext(conf=conf)
7
8 #Input file to spark read
9 text_file = sc.textFile("hdfs://master:9000/user/app/politics_news.txt")
10 counts = text_file.flatMap(lambda line: line.split(" ")) \
11     .map(lambda word: (word, 1)) \
12     .reduceByKey(lambda a, b: a + b, 1) \
13     .map(lambda (a, b): (b, a)) \
14     .sortByKey(0, 1) \
15     .map(lambda (a, b): (b, a))
16
17 #Output directory to save results
18 counts.saveAsTextFile("hdfs://master:9000/user/app/politics_count.txt")
19
```

Lista 3: Implementação da função de contagem de palavras no Spark.