UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE INSTITUTO METRÓPOLE DIGITAL

Relatório do Projeto Final da disciplina Linguagem de programação II

Tema: Sistema para detecção de fake news através de similaridade etnre conjunto de palavras

Alunos: Yan Carlos Rocha da Silva e Ubirajara Dias Viegas Júnior
Professor orientador: João Carlos Xavier Júnior

Natal 9 de junho de 2019

Introdução:

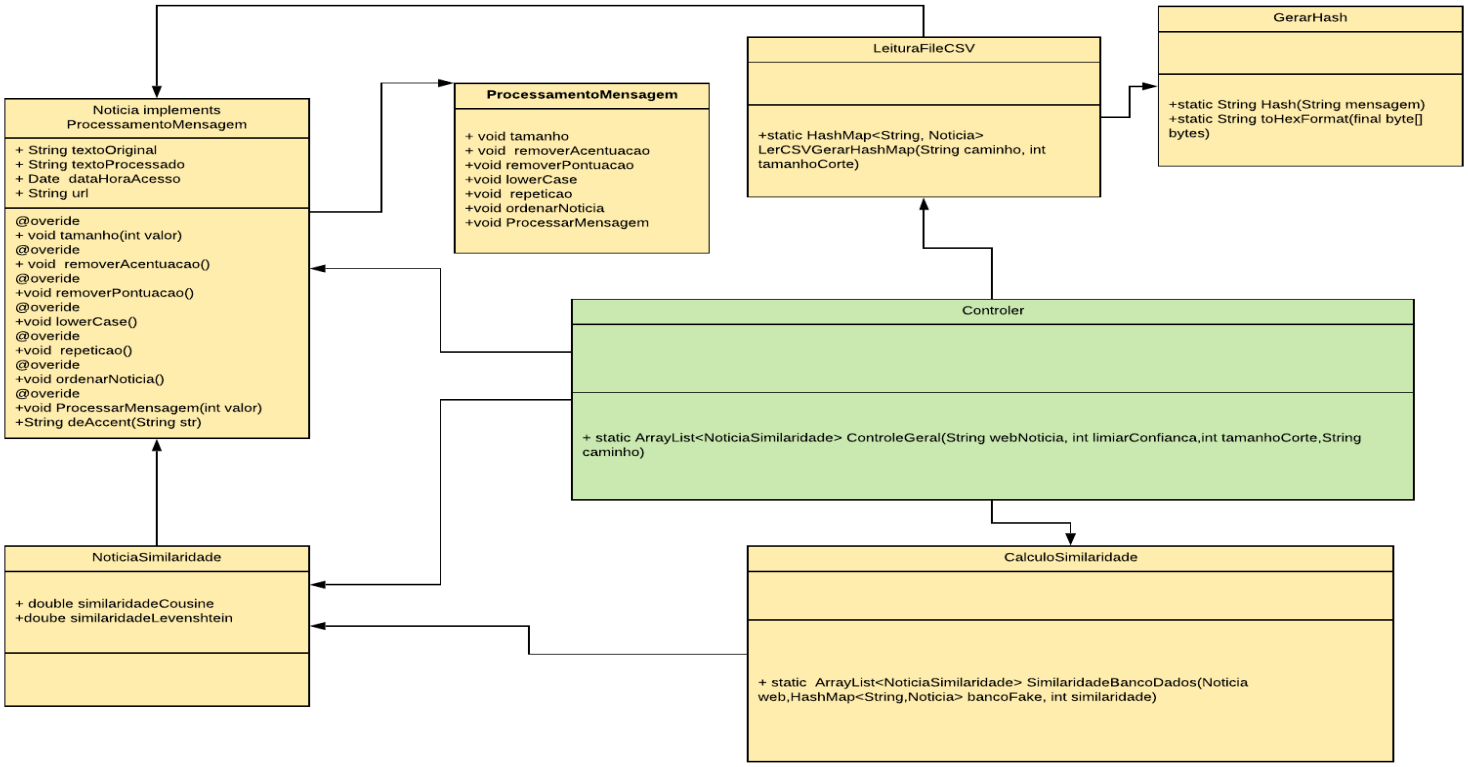
A concepção de fake news ao tratarmos sua tradução de forma literal remonta praticamente desde da existência da humanidade, afinal de contas a mentira e a transmissão de informação falsa é algo corriqueiramente antigo na história humana. Então porque hoje se dá a necessidade de se identificar e tratar uma simples mentira como um problema em larga escala? Para entendermos o fenômeno das fake news temos que traçar um paralelo com a globalização e o dinamismo do processo de transmissão de notícias hoje. Há pouco tempo se difundiu como principal mecanismo de informação os meios eletrônicos, que compreende desde as redes sociais aos grandes sites de notícia, diante desse quadro somos constantemente bombardeados por informação ao navegarmos pela internet, identificar uma notícia como falsa ou verdadeira as vezes é praticamente impossível.

Dada a dificuldade da identificação dessas notícias em um mundo cada vez mais conectado o que poderia ser apenas uma notícia falsa, ou um simples engano, toma proporções catastróficas. Podemos citar como exemplo o campo da política que recentemente vem sendo afetada por fake news, desde as eleições de Donald Trump em 2016 até o pleito eleitoral de 2018 no Brasil.

Sabendo do problema e de sua importância para com o atual momento da história, podemos denotar a importância desse projeto, que tem como objetivo central identificar fake news. O projeto consiste a partir de um banco de informações de fake news, identificar uma notícia recebida do usuário. Essa notícia será classificada como verdadeira ou falsa, dado o grau de similaridade da mesma com as notícias cadastradas no banco.

Descrição de abordagem de solução do problema:

Abaixo temos um diagrama de classes que exemplifica a dinâmica estrutural do projeto:



Na cor verde temos a classe de controle, as demais são de domínio.

* Não foi estabelecido um padrão de projeto específico para este projeto.

Descrição geral:

Classe Noticia: A classe consiste em uma estrutura capaz de armazenar de forma correta a notícia, recebendo um texto com o formato específico de Índice/Noticia para processamento/ URL/ data de acesso, é possível processar essa notícia de forma que possa posteriormente ser utilizada pelos algoritmos de similaridade em questão. A classe implementa a interface ProcessamentoMensagem, com os métodos posteriormente implementados que tem como objetivo processar a mensagem para o formato ideal.

Atributos:

String textoOriginal. (Guarda o texto original da notícia)
String textoProcessado. (Guarda o texto pós-processamento)
Date dataHoraAcesso. (Guarda a data e hora de acesso da notícia)
String Url. (Guarda a URL da notícia)

Métodos:

Métodos vigentes na interface ProcessamentoMensagem.
String deAccent(String str) (tem como objetivo remover os acentos das palavras)
Métodos gets e sets dos atributos.

Interface ProcessarMensagem: A interface tem como objetivo servir com um guia para implementação dos métodos corretos, que visam preparar a mensagem para os métodos de similaridade.

Métodos:

void tamanho (Tem como objetivo remover as palavras como tamanho inferior à valor)
void removerAcentuacao (Tem como objetivo remover a acentuação do texto)
void remvoerPontuacao (Tem como objetivo remover a pontuação do texto)
void lowerCase (Tem como objetivo deixar todas as palavras em minúsculo)
void repeticao (Tem como objetivo remover palavras repetidas)
void ordenarNoticia (Tem como objetivo ordenar a mensagem em ordem alfabética)
void processarMensagem (Invoca todos os métodos para fazer o processamento automático)

Classe LeituraFileCS: Essa classe tem como objetivo primário ler um arquivo ".csv" informado pelo usuário e armazenar o mesmo em uma estrutura HashMap, Nota: Nesta classe fazemos uso de uma API externa, a API open CSV 4.6, a mesma se encontra na pasta libs, citaremos nas referências o site da API.

Métodos:

static HashMap<String, Noticia> LerCSVGerarHashMap(String caminho, int tamanho corte) (Este método recebe como parâmetro uma String referente ao caminho, o tamanho de corte que o usuário deseja proceder no processamento da mensagem, e gera um HashMap com as notícias presentes no arquivo)

Classe GerarHash: A classe em questão tem como objetivo gerar uma chave hash no formato SHA-1 de uma string, no caso a notícia processada.

Métodos:

static String Hash(String mensagem) (Gera uma string no formato SHA-1)
static String toHexFormat(final byte[] bytes) (Apartir de uma sequência de bytes, no caso 40, faz a conversão dos mesmos para o formato string)

Classe NoticiaSimilaridade: A classe tem como objetivo armazenar uma notícia normal, já que a mesma herda de notícia, e armazenar as similaridades obtidas a partir dos testes em questão.

Atributos:

double similaridadeCosine (Armazena a similaridade de Cosine)
double similaridadeLevenshtein (Armazena a similaridade de Levenshtein)

Métodos:

Métodos gets e sets padrões.

Classe CalculoSimilaridade: A classe calcula o grau de similaridade de uma notícia com um hashMap em específico retornando um array list das notícias com maior similaridade, faz-se uso de API externa de text similarity que será devidamente citada nas referências.

Métodos:

static ArrayList<NoticiaSimilaridade> SimilaridadeBancoDados(Noticia web, HashMap<String,Noticia> bancoFake, int similaridade) (Essa classe faz uso da API de similaridade de texto, para verificar notícias similares à obtida via scraping, gerando ao final um array list com as mais similares. Faz o uso da distância de Cosine e a distância de Levenshtein para se obter o grau de similaridade)

Classe Controler: A classe tem como objetivo invocar as demais classes e gerar para interface gráfica com base nos dados fornecidos(String da notícia de scraping, int limiar de confiança, int tamanho de corte, String do caminho do arquivo) um array list com as notícias mais similares à obtida por scraping.

Métodos:

static ArrayList<NoticiaSimilaridade> ControleGeral(String webNoticia, int limiarConfianca,int tamanhoCorte,String caminho) (Tem como função gerar o array list com as notícias mais similares, funciona como um método de controle do projeto)

Conclusão:

Por meio deste projeto foi possível verificar a aplicação concreta de análise de similaridade com em um conjunto de palavras distintos, sendo possível por meio desta técnica detectar supostas fake news. Dada a importância e o impacto que essas notícias falsas tendem a gerar, principalmente nos dias de hoje, temos a real importância desse projeto.

Além dos métodos de similaridade o projeto concebeu conhecimento na área de leitura de arquivos(especificamente csv), geração de key criptográfica em SHA-1, conhecimento básico sobre web scraping, estrutura de dados HashMap, fora os conhecimentos adquiridos no decorrer da disciplina.

Referências:

Foram utilizadas duas API's nesse projeto de forma ativa, e uma terceira para complemento.

Package.org.apache.commons.text.similarity: Foi utilizada para se obter a similaridade entre Strings, através da distância de Cosine e à distância de Levenshtein

https://commons.apache.org/proper/commons-text/javadocs/api-release/org/apache/commons/text/similarity/package-summary.html

opencsv 4.6 API: Utilizada para leitura de arquivos CSV

http://opencsv.sourceforge.net/apidocs/index.html

Package org.apache.commons.lang3: Utilizada como complemento à openCSV.

https://commons.apache.org/proper/commons-lang/javadocs/api-3.1/org/apache/commons/lang3/package-summary.html

O método deAccent na classe Noticia foi obtido no link abaixo para facilitar e simplificar o processo

`https://pt.stackoverflow.com/questions/42/como-remover-acentos-e-outros-sinais-gr%C3%A1ficos-de-uma-string-em-java`