

Universidade Fernando Pessoa

Faculdade de Ciências e Tecnologias



Avaliação de Notícias

Supervisor: Paulo Rurato

Orientador: Luís Borges Gouveia

Aluno:

Vanilson Muhongo, nº 31875

Universidade Fernando Pessoa

Junho de 2020

Tabela de conteúdos

Objetivos.....	3
Qualidade dos Resultados.....	4
Arquitetura da Aplicação.....	5
Requisitos Funcionais.....	6
Requisitos Não-Funcionais.....	7
Mapa do Site.....	8
Referências.....	9

Introdução

O presente trabalho é sobre avaliação de notícias falsas, mais concretamente uma aplicação para detetar e alertar o utilizador acerca da veracidade das mesmas.

O objetivo deste trabalho é evitar que cada um de nós, sirva de caixa-de-ressonância ao partilhar e assim tornar mais credível, notícias que são falsas, fabricadas, desatualizadas, parciais e que em vez de servir para informar, servem interesses não especificados na própria notícia.

As principais tecnologias utilizadas para desenvolver esta aplicação foram Python (*Framework Flask*); Javascript; HTML e CSS; Mysql (motor de base de dados); NewsAPI (usada como motor de busca de um URL da notícia); e a API de análise de texto Microsoft (usada para retirar as palavras-chave das notícias).

Objetivos

Dado o estado da comunicação social nos tempos que decorrem e da quantidade de notícias falsas que circulam na Internet e principalmente nas redes sociais, onde habitualmente não faltam debates sobre este assunto nos comentários. O objetivo deste trabalho é o de criar uma ferramenta que proporcione uma possível referência para quem procura saber se uma determinada notícia é verdadeira ou falsa – auxiliando na determinação pessoal sobre a valorização que deve fazer desta.

Resumidamente, a aplicação desenvolvida, armazena numa base de dados, as notícias que vão sendo pesquisadas e apresenta o resultado da pesquisa feita, aos utilizadores que visitem e utilizem a aplicação.

Depois de alguma pesquisa e de me reunir com o coordenador do projeto, chegamos ao consenso de que esta seria uma solução possível para mitigar este problema, devido ao facto. Soluções mais avançadas poderiam recorrer a tecnologias como o *machine learning* ou outras técnicas associadas com a inteligência artificial. Estas abordagens constituem soluções avançadas para a criação de uma ferramenta de deteção e avaliação de notícias falsas, em especial, para o seu aprofundamento.

Qualidade dos Resultados

A ferramenta desenvolvida representa uma possível solução entre várias a um problema que infelizmente é recorrente, e que com as redes sociais tem tendência a crescer. Contudo, não dominando o uso e exploração da inteligência artificial (como a criação de *bots*) para poder comparar resultados entre várias páginas de uma forma mais eficaz, foram consideradas outras soluções.

A proposta de solução do problema seguiu a seguinte forma:

O utilizador ao introduzir um link de uma notícia no *Web site*, o título desse link e respetivo conteúdo da notícia vai ser analisado recorrendo ao serviço da *News API*, para poder indicar se uma notícia é credível ou não.

Com base numa fórmula de desenvolvimento próprio e usando um conjunto de fatores que foi possível recolher, foi decidido atribuir percentagens a cada um desses fatores de forma a ter uma percentagem de verdade sobre a notícia:

- Total de notícias devolvida pela API;
- Quantidade de notícias com mais de 30% de semelhança com a notícia:
 - que queremos avaliar – 15%
 - média dessas semelhanças – 13.3%
 - identificação do autor ou não – 18.3%
 - fonte da informação – 18.3%
 - imagem – 5%
 - data da notícia – 30%

A News API tem uma limitação diária de 1000 pedidos de pesquisa, a api não devolve a notícia na íntegra (apenas uma descrição) e nem sempre encontra a notícia pesquisada. A API da análise de texto da Microsoft tem uma limitação de utilização de 7 dias e apenas faz 5000 pedidos por mês.

Como consequência, a ferramenta desenvolvida torna-se mais lenta na análise e pesquisa da Notícia.

Arquitetura da Aplicação

A arquitetura ou modelo utilizado para desenvolver a aplicação foi o MVC que em inglês significa, *Model View and Controller*. É um padrão de arquitetura de software (*design pattern*) que separa a representação da informação da interação do utilizador com ele. O modelo consiste nos dados da aplicação, nas regras de negócios, na lógica e funções. Uma vista (*view*) pode ser qualquer saída de representação dos dados, como uma tabela ou um diagrama. É possível ter várias visões do mesmo dado, como um gráfico de barras e uma visão tabular para contadores. O controlador faz a mediação da entrada, convertendo-a em comandos para o modelo ou visão. As ideias centrais por trás do MVC são a reutilização de código e separação de conceitos.

Além de dividir a aplicação em três tipos de componentes, o MVC define as interações entre os componentes. O controlador envia comandos para o modelo para atualizar o seu estado. O controlador também pode enviar comandos para a *view* associada para alterar a apresentação da visão do modelo. Um modelo armazena dados e notifica as suas *views* e controladores associados, quando há uma mudança do seu estado. Estas notificações permitem que as *views* produzam saídas atualizadas e que os controladores alterem o conjunto de comandos disponíveis. A *view* gera uma representação dos dados presentes no modelo solicitado.

Requisitos Funcionais

Existem dois tipos de contas associadas à ferramenta desenvolvida: administrador e utilizadores.

Requisitos funcionais do **utilizador**:

- Efetuar o *login* com nome e senha
- Inserir um *link* válido de uma notícia

Nota: Estas ações poderão ser feitas através do Website.

Requisitos funcionais do **administrador**:

- Adicionar ou remover um ou vários administradores e utilizadores
- Verificar, pesquisar ou remover notícias que vão sendo inseridas na base de dados

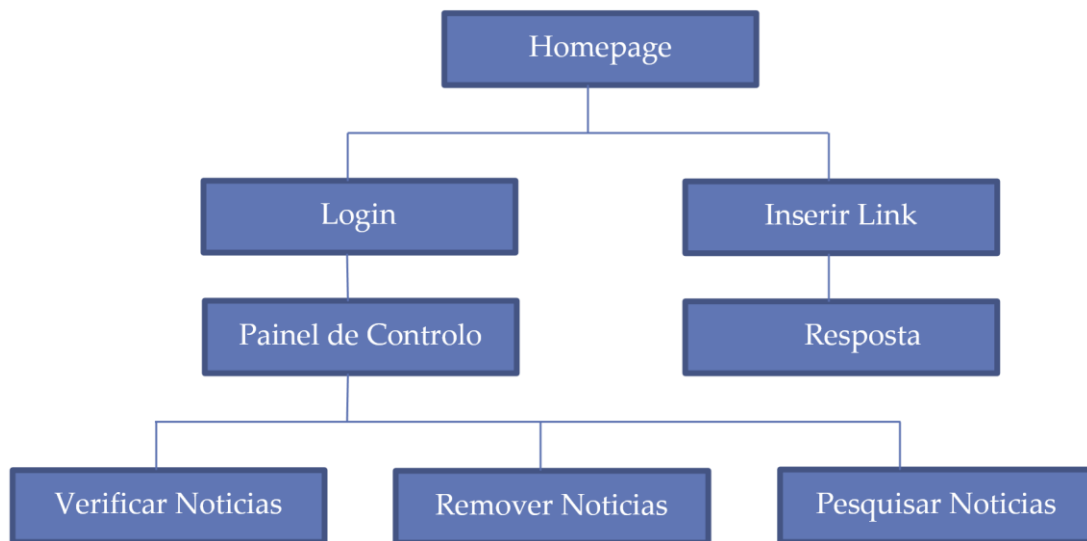
Requisitos Não-Funcionais

São requisitos não funcionais para a ferramenta:

- Para aceder à aplicação, é necessário que o utilizador esteja conectado à rede
- O padrão de arquitetura de software utilizado será MVC com recurso à *framework Flask*
- A aplicação deverá ser fácil de usar e que dispense muitos recursos gráficos
- A base de dados deve ser protegida para acesso apenas a utilizadores autorizados, neste caso aos administradores
- A aplicação deverá garantir que não haja perda de dados, e deve ser capaz de tratar exceções e de recuperar de falhas
- A interface da aplicação deverá comporta-se adequadamente independente do *front-end* que será utilizado para acesso – Browser, Smartphone ou Tablet
- Ligação à Internet/Web para uso de API
- Optimização da API

Mapa do Site

Website:



O mapa da ferramenta mostra o top-down associado com as suas funcionalidades. Com base na estrutura, percebe-se a sua simplicidade. O objetivo é colecionar notícias que tenham sido avaliadas, com base nos respetivos URLs. Para tal, o utilizador tem de estar registado e gerir as notícias que submeteu e mantém numa base de dados (de notícias) pessoal, que pode pesquisar.

Referências

São listados as fontes para suporte do desenvolvimento conceptual da ferramenta.

- 1 – <http://www.fakenewschallenge.org/>
- 2 – <https://towardsdatascience.com/detecting-fake-news-with-deep-learning-7505874d6ac5>
- 3 – <https://www.mindtools.com/pages/article/newstml106.html>
- 4 – <https://www.torontopubliclibrary.ca/spotfakenews/>
- 5 – <https://azure.microsoft.com/pt-pt/services/cognitive-services/text-analytics/>