UNIVERSIDADE
LUSÓFONA

Utilização da Blockchain para combater a propagação de fake-news

Filipe Cacho | 21702361

Trabalho Final de Curso | LEI | 21/07/2023

Orientado por: João Carvalho e Luís Gomes

www.ulusofona.pt

Agradecimentos

Agradeço ao professor Luís Gomes e ao professor João Carvalho por me terem orientado ao longo deste projeto, por estarem sempre atentos ao desenvolvimento ao longo deste projeto de investigação e por estarem constante disponíveis.

Resumo

Fake-news é cada vez mais um tópico de maior relevância na sociedade atual, devido ao facto que cada vez mais pessoas possuem acesso à Internet, grandes organizações e indivíduos aperceberam-se há algum tempo que a maioria das pessoas não verifica a veracidade das notícias que leem, o que permite a criação de modelos e ferramentas destinados a criar noticias falsas com o objetivo de desacreditar factos inconvenientes e até criar mentiras sobre os mais variados tópicos.

Cada vez mais, todos os dias, as pessoas estão a ser vítimas de notícias falsas, desde o individuo que está a navegar no Reddit e lê uma notícia falsa ou até aos milhares de pessoas que leem noticias falsas sobre pessoas com relevância nacional, sendo a maioria dos casos com o intuito de denegrir tal pessoa.

Um bom exemplo do perigo das fake-news seriam as eleições presidenciais de 2016 dos Estados Unidos, aonde as Fake-news foram uma ferramenta usada para espalhar falsas informações nas redes sociais, havendo fortes indícios de intervenção de atores externos no processo democrático.

Fica claro que é necessária uma forma de permitir às pessoas verificar rapidamente se as notícias que estão a ler são baseadas em factos, reais para permitir que as pessoas criem as suas opiniões pelo menos utilizando informações corretas.

Para tal, fica proposto neste projeto a criação de uma Blockchain com o objetivo de criar um local para armazenar noticias de fontes consideradas reputáveis para que depois se possam comparar com notícias que o utilizador está a ler para se tentar perceber se o conteúdo é baseado em informação verdadeira ou não utilizando algoritmos de processamento de distância métrica.

Abstract

Due to the increasing number of people with internet access, fake news has become a prevalent and concerning issue. It has been observed that a majority of individuals do not fact-check the news they consume. This situation has led to the development of models and tools specifically designed to create news that contradicts established facts or spreads deliberate falsehoods across various topics.

Every day, more and more people fall victim to fake news. Whether it's someone browsing through Reddit and stumbling upon a fabricated article or the thousands who encounter false narratives about prominent individuals, the primary objective often revolves around defaming the person in question.

The 2016 USA presidential elections serve as a prime example of the dangers posed by fake news. During that time, false information was deliberately disseminated through social networks, and there is substantial evidence pointing towards external actors attempting to manipulate the democratic process.

It becomes evident that there is a pressing need to provide people with a reliable means of quickly verifying the factual basis of the news they encounter. This project aims to address this issue by leveraging Blockchain technology to create a repository of trusted news sources. This repository can then be used to compare and validate the content of news articles, enabling users to make informed decisions based on accurate information.

To achieve this, natural language processing algorithms will be employed to analyze the language used in the news articles and determine their credibility.

Índice de Conteúdos

Capítulo 1. Identificação do problema 1
Capítulo 2. Viabilidade e pertinência 5
Capítulo 3. Benchmarking 9
Capítulo 3.1 Objetivos concretos do projeto 12
Capítulo 4. Engenharia 18
Capítulo 4.1 Levantamento e análise de requisitos 19
Capítulo 4.2 Cenários de aplicação 21
Capítulo 4.3 Diagramas 23
Capítulo 5. Solução proposta 27
Capítulo 5.1. Arquitetura 27
Capítulo 5.2. Tecnologias, ferramentas utilizadas e fundamentação 31
Capítulo 5.3. Implementação 36
Capítulo 5.4. Abrangência 44
Capítulo 5.4.1. Limites do projeto 45
Capítulo 5.4.2. Continuidade do projeto 46
Capítulo 6. Método e Planeamento 47
Capítulo 7. Resultados 49
Capítulo 7.1. Resultados dos algoritmos implementados 49
Bibliography 55

Tabela de Figuras

Figura 1 - Donald Trump Clickbait [2] 1
Figura 2 - Declaração falsas no Twitter [2] 2
Figura 3 - Falsa informação sobre as regras de votação [1] 2
Figura 4 - Noticia com título enganador [1] 3
Figura 5 - Noticia com declarações falsas [2] 3
Figura 6 - Classificação manual dos artigos 6
Figura 7 - gráfico com 1137 artigos classificados 7
Figura 8 - Comparar artigo do utilizador 8
Figura 9 - software para auxiliar a vacinação mundial 9
Figura 10 - Logótipo da News Provenance Project [6] 10
Figura 11 - Logótipo da Safe.press [7] 10
Figura 12 - Logótipo da Deeptrust Alliance [8] 10
Figura 13 - Selo de qualidade ANSI 11
Figura 14 - Esquema da blockchain FACT 12
Figura 15 - Elementos mantidos e descartados pela Blockchain proposta 16
Figura 16 - Principais tecnologias usadas no projeto 19
Figura 17 - Extração de artigos 24
Figura 18 - Criação dos blocos 25
Figura 19 - Verificar veracidade dos artigos 25
Figura 20 - Cálculo das novas categorias 26
Figura 21 - Dependência dos módulos do mongoDB 28
Figura 22 - Processo de criação de blocos 28
Figura 23 - Representação da Blockchain [16] 34
Figura 24 - Exemplo de uso da Blockchain [17] 35
Figura 25 - Representação da maioria dos nodes (52%) [18] 35
Figura 26 - extrair URLS 37
Figura 27 - Remover lixo 37
Figura 28 - Remover duplicados 38
Figura 29 - Extrair partes do artigo 39
Figura 30 - Detetar linguagem do corpo do artigo 39
Figura 31 - Prova de trabalho 40
Figura 32 – Excerto do ficheiro .json da blockchain 41
Figura 33 - classificação de artigos 41
Figura 34 - Divisão dos artigos em 30% e 70% 42
Figura 35 - cálculo da categoria usando similaridade de cosseno 43
Figura 36 - gráfico com as categorias originais 43
Figura 37 - comparar artigo do utilizador com a base de dados blockchain 44
Figura 38 - Cronograma do projeto 47
Figura 39 - Resultado não pretendido caso 1 51
Figura 40 - Resultado pretendido caso 1 51

Capítulo 1. Identificação do problema

Noticias falsas (fake-news) é um termo que ganhou maior relevância nos últimos anos devido ao fácil acesso à internet por um número cada vez maior de pessoas no mundo [1].

Normalmente as pessoas obtém as notícias a partir de fontes de informação consideradas fidedignas como jornais, canais de televisão, etc. uma vez que estes tem de seguir códigos de conduta e ética bastante rigorosos, mas que ainda assim podem cometer erros nos processos de verificação dos factos. No entanto a internet permite a qualquer pessoa publicar qualquer tipo de notícia que não tem de obedecer a regras. Começa a ser cada vez mais a norma as pessoas obterem as notícias a partir de redes sociais e sites que aparecem nas páginas que visitam.

Existem alguns tipos de notícias falsas:

-Clickbait, são notícias montadas com a intenção de atrair tráfego ao site, são geralmente tópicos que apelam à maioria da população cujo único objetivo é aumentar os lucros do site [1].

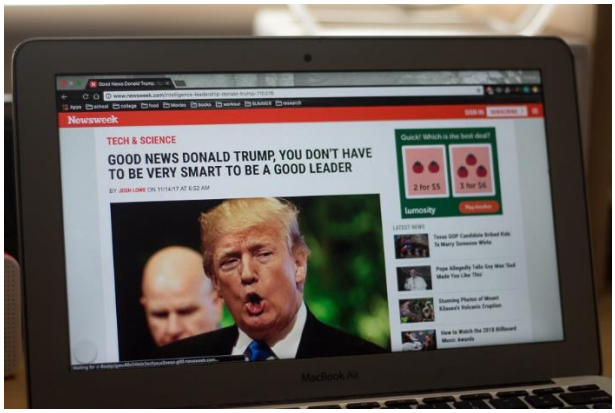


Figura 1 - Donald Trump Clickbait [2]

-Propaganda, são notícias criadas para enganar as audiências distorcendo os factos para influenciar a opinião das pessoas para ganho próprio.

Figura 2 - Declaração falsas no Twitter [2]

-Não verificadas, às vezes jornalistas publicam noticias sem terem verificado todos os factos o que pode enganar a audiência

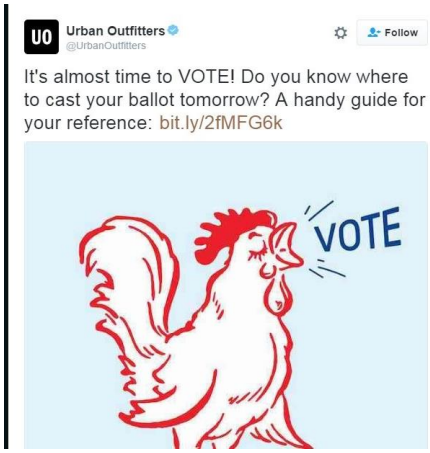


Figura 3 - Falsa informação sobre as regras de votação [1]

-Títulos falsos, são notícias que apesar do conteúdo ter sido verificado e é verdadeiro, o título é escrito de forma incoerente e engana o espectador. Este tipo de notícias espalha-se rapidamente pelas redes sociais e sites aonde se vê apenas o título.

Wednesday, May 15, 2013 www.dercotimes.com DAILY TIMES / 21

SPECIAL ADVERTISING FEATURE ©2013 UNIVERSAL COMMERCE 8000 FREEDOM AVE., N. CANTON OH 44720

Public gets Free TV with no monthly bills

Federal law makes TV network giants broadcast Free TV signals regionally in crystal clear digital picture in all 50 states allowing U.S. households to pull in Free TV with a sleek $49 micro antenna device engineered to pull in nothing but Free TV channels with no cable, satellite or internet connection and no monthly bills

Who Gets Free TV: Listed below are the Primos area zip codes that can get Free over the air TV channels. If you find the first two digits of your zip code immediately call: **1-888-752-7147**

Figura 4 - Noticia com título enganador [1]

-Noticias que confirmam as nossas opiniões (biassed), são notícias que confirmam o nosso ponto de vista e muitas vezes as pessoas procuram notícias do género sem prestar atenção à autenticidade da fonte.

Pope Francis Shocks World, Endorses Donald Trump for President, Releases Statement

TOPICS: Pope Francis Endorses Donald Trump

Figura 5 - Noticia com declarações falsas [2]

Fica portanto aqui claro que a proliferação de notícias falsas é cada vez mais comum, as pessoas no seu dia-a-dia não tem tempo e não estão interessadas em verificar a autenticidade das notícias que leem, preferem consumir o conteúdo que a Internet lhes oferece sem ter que pensar nestes aspetos, o que leva a que formem opiniões sobre factos potencialmente errados e que poderão transmitir aos que os rodeiam [3].

As notícias falsas são consumidas facilmente porque são sensacionalistas e quando chegam ao publico alvo certo, estes encarregam-se muitas vezes de as enviar para os seus contactos.

É de extrema necessidade a criação de uma ou mais ferramentas que permitam às pessoas facilmente perceber se as notícias que estão a ler são verdadeiras ou não.

Capítulo 2. Viabilidade e pertinência

No início do desenvolvimento deste projeto, foi proposto um programa que poderia ajudar a combater as notícias falsas utilizando uma combinação de uma estrutura do tipo blockchain para armazenar os dados recolhidos de websites com boa reputação jornalística.

Posteriormente, esses artigos seriam analisados por algoritmos capazes de calcular matematicamente a semelhança entre 1 e outros tantos artigos e devolver o mais semelhante. O utilizador poderia inserir o URL de um artigo à sua escolha, para que este fosse devidamente processado e o corpo do texto fosse comparado com os artigos já pré-processados na base de dados, e o programa devolveria ao utilizador uma resposta sobre o quão semelhante o artigo é em relação aos artigos armazenados.

Logo no início do projeto, foi decidido que seria aplicada a lógica de absolutismo informacional. Ou seja, assumiu-se que os artigos consumidos das fontes selecionadas estariam corretos e que as informações foram devidamente verificadas pelos responsáveis pela publicação. Isso, porque seria impraticável e até mesmo impossível criar um sistema paralelo para verificar a autenticidade dos artigos.

Durante a fase inicial do projeto, estava previsto alimentar a base de dados com artigos de vários websites. No entanto, rapidamente ficou claro que cada website de notícias armazena as informações dos artigos de forma completamente diferente, tornando difícil extrair apenas as informações relevantes de todos os websites. Considerando a sensibilidade dos algoritmos de distância métrica, arriscar ter artigos com informações irrelevantes na base de dados não seria ideal. Por isso, foi necessário criar um módulo específico para a extração de notícias. Devido aos inúmeros testes necessários, apenas foi possível implementar a extração dos artigos do site https://apnews.com/

Ao longo do desenvolvimento, tornou-se evidente que o projeto tinha um caráter mais científico e ultrapassou o escopo inicialmente planeado. Com isto o projeto sofreu uma transformação mas não perdeu viabilidade, tendo passado de uma solução mais do âmbito comercial a um projeto científico que estuda a fiabilidade dos algoritmos de distâncias métricas como uma possível forma de combater as notícias falsas.

Para comprovar a fiabilidade desses algoritmos, foi necessário implementar um sistema de classificação dos artigos em 10 categorias, o que não estava inicialmente previsto. Esse sistema exige que o utilizador classifique manualmente os artigos, uma vez que a criação de um sistema totalmente automático e com um alto grau de fiabilidade de classificação seria um projeto separado só por si mesmo.

A imagem abaixo mostra o sistema de classificação manual dos artigos do programa.

Figura 6 - Classificação manual dos artigos

O propósito desse sistema é permitir ao utilizador verificar se os 4 algoritmos de distância métrica implementados (**similaridade do cosseno, distância angular, distância euclidiana e distância de Minkowski**) são capazes de calcular corretamente as categorias dos artigos, baseando-se nos artigos já classificados pelo utilizador.

O trabalho manual do utilizador permite aferir o grau de fiabilidade desses algoritmos. Após a classificação, o projeto divide os artigos em dois conjuntos: 30% para testes e os restantes 70% para treino. O projeto calcula a categoria para os artigos no conjunto dos 30% utilizando o conjunto dos 70% como base de treino e considera um artigo bem classificado quando a categoria calculada é igual à categoria atribuída pelo utilizador.

Caso o utilizador não queira classificar ele próprio os artigos disponíveis na base de dados blockchain, é disponibilizada uma base de dados já pré-classificada com 1137 artigos para que o utilizador possa analisar os resultados.

A imagem abaixo mostra o gráfico interativo com os resultados da base de dados com 1137 artigos já classificados.

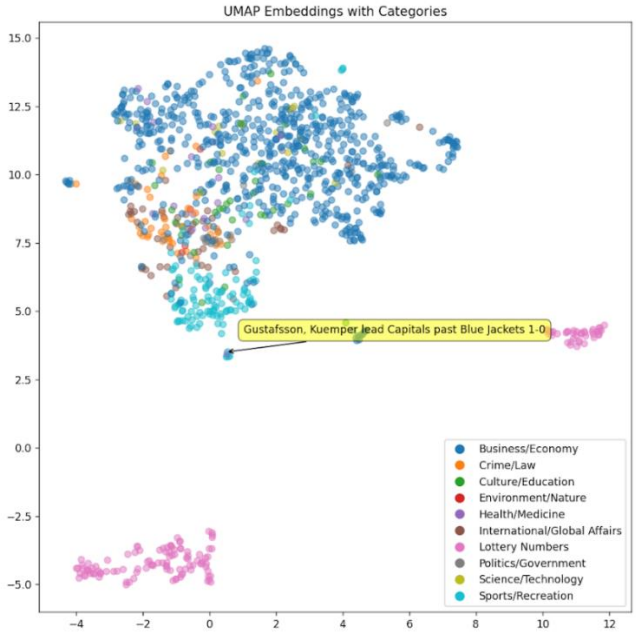


Figura 7 - gráfico com 1137 artigos classificados

Todo o processo manual descrito acima, serviu como 1º teste no projeto para estudar a fiabilidade dos algoritmos de distância métricas implementados. Ao construir um sistema de recategorizarão dos artigos utilizando estes algoritmos, é possível de se observar o seu nível de fiabilidade ao estudar a percentagem de categorias devidamente calculadas. O código aqui implementado, foi praticamente todo reaproveitado para a implementação da componente abaixo descrita, que permite comparar 1 artigo do utilizador com todos os artigos armazenados na base de dados.

Todo o processo descrito acima de classificação manual dos artigos serviu de base para o derradeiro teste, que é permitir ao utilizador inserir o corpo de um artigo qualquer à sua escolha e compará-lo com todos os artigos já armazenados. Estava inicialmente previsto que o utilizador colocasse apenas o URL do artigo pretendido e que o corpo do artigo fosse automaticamente extraído, mas tal não é possível por cada site guarda os artigos de forma diferente e portanto não é possível garantir a extração correta do corpo do artigo sem lixo, não deixando outra alternativa se não pedir ao utilizador que selecione ele mesmo o corpo do artigo.

A imagem abaixo mostra o resultado do programa quando se insere o corpo do artigo https://www.pbs.org/newshour/politics/supreme-court-unanimously-rules-for-deaf-student-in-education-case que é sobre o caso de um estudante surdo, e o programa devolve como sendo o artigo mais semelhante que encontra o https://apnews.com/article/michigan-state-government-education-d10553f4ea6f73eef5585427de0fe370 que é de facto o artigo que o "Apnews" tem sobre este caso e que existia na minha base de dados, sendo este o resultado esperado.

O programa apresenta um score = 0.8, quanto mais semelhantes os artigos são mais alto será o valor do score, sendo o máximo score = 1.

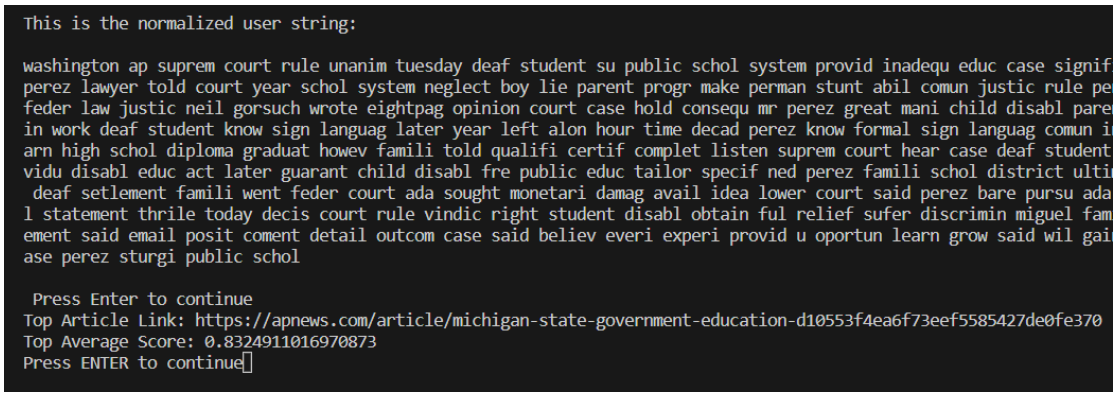


Figura 8 - Comparar artigo do utilizador

Fica assim aqui resumido o objetivo do projeto e o produto final obtido, a pertinência deste projeto é elevada porque pretende dar o seu contributo para a resolução de um problema que a sociedade atual enfrenta e com os novos avanços no ramo da inteligência artificial vai ficar cada vez mais difícil distinguir noticias falsas.

O desenvolvimento de projetos deste género vão permitir ao publico geral filtrar com facilidade noticias falsas e manterem-se a par do que está a acontecer no mundo, cada vez mais o jornalismo sensacionalista tira acontecimentos e afirmações fora de contexto e provoca reações de revolta por parte da sociedade geral, o que aumenta ainda mais a necessidade de projetos que queiram contribuir para a resolução deste problema.

Capítulo 3. Benchmarking

Há muito que a blockchain deixou de ser uma estrutura dedicada única e exclusivamente às cripto-moedas, já existem milhares de projetos que alargaram os horizontes.

Desde software que utiliza a blockchain para auxiliar à vacinação a uma escala internacional, promovendo assim a cooperação entre diferentes instituições a nível mundial utilizando um registo descentralizado e com latências muito baixas [4].

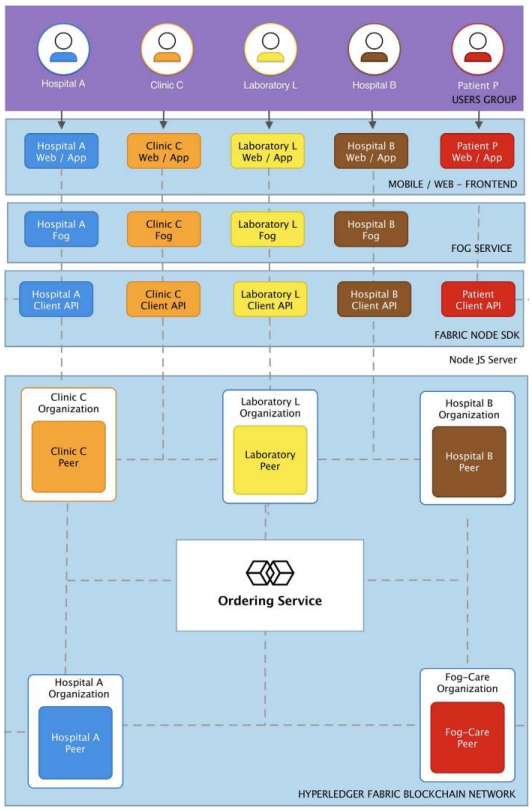


Figura 9 - software para auxiliar a vacinação mundial

Já existem vários projetos que usam a Blockchain como ferramenta de combate às notícias falsas, mas a maioria deles concentra-se na deteção de imagens de notícias que foram editadas para distorcer o seu conteúdo e serem usadas fora de contexto.

Um exemplo de tal projeto é o "News Provenance Project", que pretende guardar as imagens publicadas nos media e manter um registo imutável da sua origem para consulta. Infelizmente não existe documentação sobre como a Blockchain interage com os algoritmos descritos e implementados [5].

Figura 10 - Logótipo da News Provenance Project [6]

Outro projeto semelhante é o "Safe Press" da IBM que pretende adicionar um selo digital aos artigos considerados de confiança e guardá-los na Blockchain [7], mas infelizmente não há praticamente informação disponível sobre como é que o projeto funciona.

Figura 11 - Logótipo da Safe.press [7]

O "DeepTrust Alliance" é outro projeto que procura combater as manipulações que usam inteligência artificial para gerar vídeos a personificar pessoas ao vivo (deep-fakes), sendo o objetivo desta organização utilizar a Blockchain juntamente com inteligência artificial para identificar estes vídeos [6]. Mas infelizmente não é possível encontrar nenhuma informação sobre como é que a tecnologia que propõe funciona.

Figura 12 - Logótipo da Deeptrust Alliance [8]

Um outro projeto que usa a blockchain para detetar noticias falsas é o "ANSAcheck", que consiste na utilização da blockchain para guardar a origem das notícias e depois atribuir um selo de qualidade a esse artigo porque o mesmo se encontra na Blockchain e, portanto, a sua origem pode ser verificada. Segundo os autores, isto permite ao leitor um elevado nível de confiança porque a origem das notícias pode ser verificada [9].

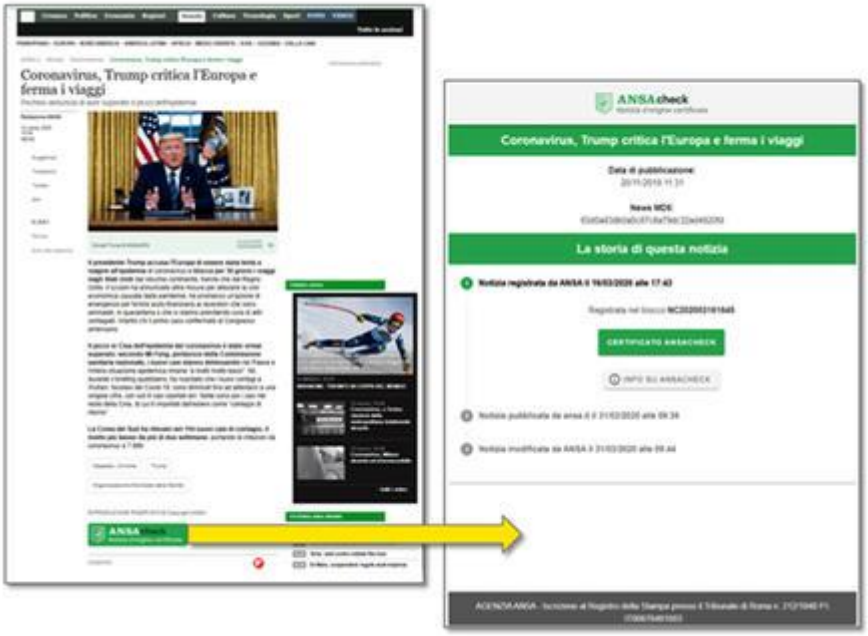


Figura 13 - Selo de qualidade ANSI

Um dos projetos mais interessantes e que explica exatamente como funciona é o “Fact Protocol” (FACT). É um projeto que consiste em montar uma blockchain para que consiste em 2 grupos de pessoas:

- Pessoas que registam as notícias, são pessoas que recebem tokens como incentivo por publicarem as notícias.

- Os validadores, são pessoas que fazem verificam os factos apresentados pelo grupo de pessoas descritos acima e que votam positivamente ou negativamente contra uma notícia.

Caso os votos sejam maioritariamente positivos, a pessoa que colocou a notícia recebe tokens da plataforma. As pessoas que fizerem a verificação dos factos também recebem tokens [10].

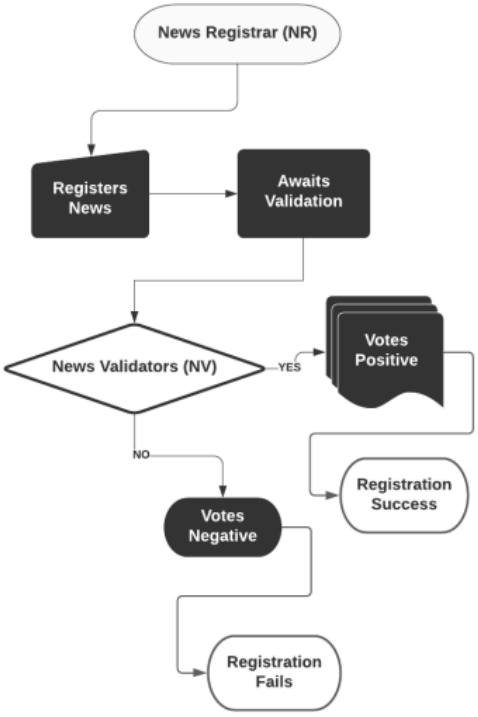


Figura 14 - Esquema da blockchain FACT

O documento oficial descreve os múltiplos processos disponíveis, como por exemplo alguém se pode tornar validador, ou como apelar caso sejam atribuídos votos negativos a um artigo quando o autor considera este ser verdadeiro.

Apesar de já existirem várias tentativas de utilizar a Blockchain para combater as notícias falsas, nenhum deles pretende fazer exatamente o mesmo a que este projeto se propõe.

Capítulo 3.1 Objetivos concretos do projeto

No capítulo acima foram apresentados vários exemplos de outros projetos que utilizam a blockchain para auxiliar os mais variados setores (como por exemplo a saúde) mas o foco do capítulo do benchmarking foi mesmo projetos relacionados a noticias falsas.

Grande parte dos projetos que pretendem ajudar a resolver os problemas devido a noticias falsas, pretendem atacar a manipulação e distribuição de imagens falsas através do uso de algoritmos proprietários, estes são mantidos em segredo para não serem desenvolvidas formas de serem ultrapassados rapidamente e para serem utilizados para fins comerciais.

Como este projeto não está relacionado de modo nenhum com processamento de imagens, esses tipos de projetos são paralelos a este e não uma competição direta.

O outro grande grupo de projetos que existe relativos ao uso da blockchain para combater noticias falsas, são projetos como “ANSAcheck” e o “safe.press” que segundo a informação disponível sobre eles e resumindo simplificadamente, consistem em colocar um selo de qualidade nos artigos publicados pelos seus parceiros. Utilizando esse selo o leitor pode seguir a notícia até à sua fonte porque a mesma está armazenada numa blockchain. Estes tipos de projetos também são paralelos a este projeto, uma vez a blockchain deles confia nos seus parceiros e oferece um selo de qualidade automático sem fazer nenhuma verificação.

Por fim temos o “FACT” que tenta combater as notícias falsas incentivando as notícias a serem verificadas por indivíduos em troca de tokens. Podem surgir inúmeros problemas com um sistema desses, aonde por exemplo um grande grupo de indivíduos se organiza e combina com os outros para só votarem as notícias de modo a gerarem o máximo de recompensas por exemplo. Este projeto também é paralelo ao que está aqui a ser desenvolvido, porque o objetivo é tirar o máximo possível o elemento humano do controlo e criar uma solução que no futuro seria descentralizada e automática, o que é totalmente o oposto do FACT.

Portanto este projeto propõe o seguinte:

- Utilizar a estrutura da Blockchain programando uma de raiz, para criar um repositório de artigos considerados de origem fidedigna e cujos autores não deixam a sua opinião distorcer os factos. O objetivo deste projeto não é avaliar a origem das fontes selecionadas, para tal seria preciso uma equipa de investigação jornalística, como tal não é possível, temos de partir de algum lado, e, portanto, são selecionadas fontes de artigos de excelência como base.

- Processar os artigos e extrair o seu conteúdo para armazenar no projeto. Todos os sites guardam as notícias de forma diferente no seu html. Para evitar extrair conteúdo a mais e desnecessário é preciso construir um modulo capaz de extrair concretamente o que precisamos e muito provavelmente cada site vai precisar de um modulo diferente, uma vez que não existe uma solução para todos os casos.

- Guardar os artigos extraídos em ficheiros e guardar a sua hash na blockchain. O objetivo é assim criar um sistema que se fosse acompanhado de criptografia avançada, métodos de autenticação, etc. seria praticamente impossível de falsificar, porque ao mudar um único caractere num dos artigos ou em qualquer parte da blockchain faz com que as verificações falhem. Para este projeto, questões como login e descentralização não podem ser implementadas devido ao limite de tempo, mas fica como proposta futura. A única encriptação que vai ser implementada será as hashes da blockchain e do conteúdo dos artigos e as devidas verificações para garantir que a blockchain não foi modificada.

- O conteúdo da blockchain deve ser publico. Vai contra o espírito do projeto e do combate às notícias falsas se a blockchain for privada. Se assim o fosse, o utilizador só tinha mais razões para desconfiar do sistema do que se ele for totalmente aberto.

- O utilizador deve conseguir inserir um artigo para comparar com os artigos guardados. É pretendido que o programa receba artigos do utilizador para que este possa verificar a sua autenticidade. Assumindo que o programa consegue responder se o artigo inserido é verdadeiro ou não, o utilizador pode consultar a blockchain e ver que artigo apoia ou contraria o artigo inserido uma vez que a blockchain é publica. O projeto considera que o artigo é verdadeiro quando o resultado devolvido é um artigo da base de dados blockchain que é sobre o mesmo tópico. Cabe ao utilizador decidir se o artigo devolvido é de facto coerente com o corpo do artigo que inseriu. Só a capacidade de detetar se o artigo devolvido é o coerente com o do utilizador seria outro projeto só por si mesmo.

- **Não existe tokens ou sistema de recompensas.** Não é o objetivo do projeto incentivar à publicação de artigos e verificação por pessoas em troca de recompensas. A verificação deverá ser o mais autónoma possível de modo a eliminar o erro humano. Tratando-se de um protótipo o método de prova de trabalho para gerar novos blocos é propositadamente simples para não obrigar a grandes cálculos computacionais. A fórmula para gerar novos blocos tem sempre de existir e a sua complexidade pode ser ajustada consoante o necessário.

- A comparação com os artigos inseridos deverá ser feita utilizando um ou mais algoritmos de distância métrica. Não é o objetivo do projeto inventar um novo algoritmo de combate às notícias falsas, mas sim utilizar os algoritmos disponíveis na internet para verificar a veracidade entre o conteúdo da blockchain e o artigo inserido. Parte do desenvolvimento do projeto consiste na adaptação destes algoritmos aos cenários

previstos no projeto.

Em resumo o projeto destaca-se dos outros referidos porque:

- Porque não pretende atribuir um selo de qualidade aos artigos, mas sim construir um repositório público utilizando a estrutura da blockchain reconhecida pela sua segurança.

- Não pretende dar incentivos ou processar imagens, apenas texto.

- Não pretende esconder o seu registo porque o mesmo é publico.

- Não pretende inventar nenhum algoritmo de processamento para combate às noticias falsas, mas sim testar os algoritmos de distância métrica selecionados de modo a verificar se estes podem ajudar no combate às noticias falsas e por fim permite ao utilizador verificar o conteúdo que lê, coisa que mais nenhum projeto ambiciona fazer, mas sempre dentro dos limites do repositório de notícias que o programa tem.

O esquema abaixo mostra que informações serão guardados nos blocos da Blockchain sempre que um artigo é adicionado. Dados como o Autor, Título, site aonde o artigo se encontra e outros são guardados todos num bloco.

Por sua vez dados como imagens, links externos e outros são descartados e não são guardados na Blockchain porque não são necessários para a analise da notícia.

Para tal é necessário um módulo que extraia o conteúdo todo da notícia da web e seja capaz de separar o conteúdo a manter do conteúdo a descartar.

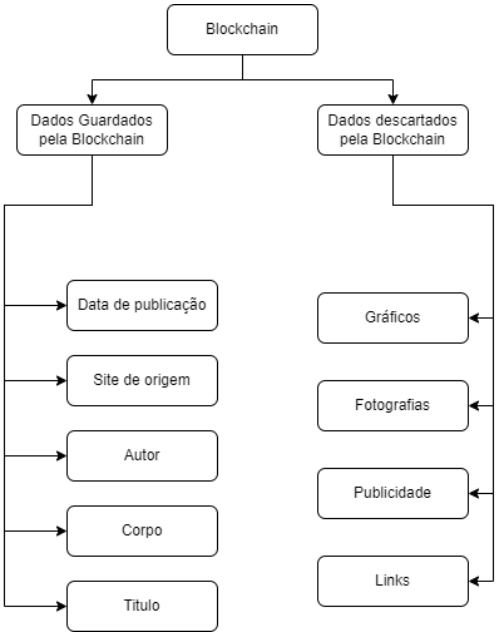


Figura 15 - Elementos mantidos e descartados pela Blockchain proposta

Sendo este um projeto do fórum académico, fica assim demarcado dos outros projetos que tentam contribuir para o combate às notícias falsas. Todos os outros projetos comerciais mencionados utilizam focam-se sobretudo na deteção de imagens e na atribuição de selos de qualidade a artigos. No entanto este projeto estuda e demonstra a possível fiabilidade da utilização da combinação de algoritmos de distância métrica juntamente com uma base de dados que utiliza um sistema de hashes seguro para garantir a integridade da mesma e faz algo que mais nenhum projeto tenta fazer: Permitir ao utilizador inserir um artigo e indicar se este é verdadeiro ou não ao comparar a composição do texto inserido com a conteúdo da base de dados de artigos já extraídos.

O projeto considera que o artigo é verdadeiro quando o resultado devolvido é um artigo da base de dados blockchain que é sobre o mesmo tópico. Cabe ao utilizador decidir se o artigo devolvido é de facto coerente com o corpo do artigo que inseriu. Só a capacidade de detetar se o artigo devolvido é o coerente com o do utilizador seria outro projeto só por si mesmo.

Estando o desenvolvimento do projeto concluído, os principais objetivos propostos no inicio do TFC foram todos alcançados, apesar de ter sido necessário acrescentar módulos não inicialmente considerados no código que demonstram a fiabilidade dos algoritmos escolhidos através de gráficos e de divisão dos dados em 30% de teste e 70% para treino, a proposta inicial do projeto era uma base de dados blockchain acoplada a um sistema

que fosse capaz de informar o utilizador se o artigo que ele consultou é verdadeiro ou não, ambos os objetivos foram definitivamente alcançados.

Para mais a viabilidade do TFC não fica de modo nenhum esgotada. Podem ser adicionados módulos adicionais para extrair mais artigos de outros websites, pode ser adicionado uma interface gráfica com divisão de permissões de administrador e utilizador normal, podem ser adicionados ainda mais algoritmos que complementem os já existentes e até pode ser adicionado algoritmos de inteligência artificial para tornar o projeto ainda mais robusto.

Capítulo 4. Engenharia

O objetivo deste capítulo será descrever os requisitos de software para o projeto descrito nos capítulos anteriores. Os requisitos deste capítulo irão permitir perceber com mais detalhe as capacidades funcionais e técnicas do projeto assim como os seus requisitos não funcionais.

Abaixo encontra-se o diagrama das principais tecnologias utilizadas no TFC e como elas se relacionam, sendo as principais tecnologias usadas as seguintes:

- Python, é a linguagem de programação de todo o projeto, é utilizada para extrair os artigos dos sites, criar a Blockchain e executar os algoritmos de processamento de linguagens naturais.

- Blockchain, é a estrutura de dados mais importante do projeto, todos os artigos extraídos são inseridos nesta estrutura que é à prova de alterações externas, para garantir tal funcionalidade, é gerada uma cifra da BD após a mesma ser chamada, essa cifra é guardada num ficheiro na pasta do projeto e é usada para verificar se a BD foi alterada cada vez que é chamada.

Quando são adicionados outros artigos à BD, é comparada a cifra da BD atual que é gerada no momento e esta é comparada com a cifra do ficheiro. Se as duas não forem iguais então a BD foi alterada e será descartada para forçar que seja feito o download dos artigos outra vez.

- MongoDB, é a base de dados de escolha para armazenar a blockchain. Foi escolhido em vez do SQL por ser mais flexível, escala horizontalmente, ou seja, consegue aguentar com grandes quantidades de dados facilmente. É de alta performance. Usa uma estrutura em documento o que a torna uma escolha melhor para guardar tipos de dados complexos.

- Algoritmos de distância métrica, é o conjunto de todos os algoritmos que são usados para comparar o artigo que o utilizador insere no programa com os artigos da blockchain um a um. É com estes algoritmos que vamos conseguir encontrar o artigo mais semelhante caso este exista na blockchain.

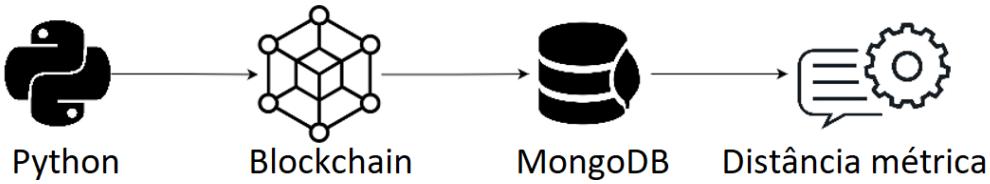


Figura 16 - Principais tecnologias usadas no projeto

Capítulo 4.1 Levantamento e análise de requisitos

A lista abaixo apresenta os requisitos técnicos e funcionais que representam de modo geral todas as funcionalidades do programa. Tendo o projeto sido completado com sucesso, todos os requisitos da tabela que se encontra abaixo foram todos implementados com sucesso

Categoria	**Descrição**
Requisito funcional	O software deverá ser capaz de extrair um ou mais artigos de uma lista de sites selecionados
Requisito funcional	O software deverá analisar e extrair os artigos selecionados para remover os hyperlinks, imagens, gráficos e outras informações não uteis, mantendo apenas o título, a data de publicação, o autor e o corpo da notícia
Requisito técnico	A informação extraída deverá poder ser guardada num ficheiro de .json
Requisito funcional	O ficheiro .json gerado deverá ser mostrar o conteúdo de todos os blocos da blockchain
Requisito técnico	O programa deverá ser capaz de detetar quando um bloco da blockchain foi alterado e, portanto comprometida
Requisito funcional	O programa deverá utilizar algoritmos de distância métrica para comparar o texto dos artigos armazenados na blockchain com um corpo de um artigo inserido pelo utilizador e determinar a veracidade do mesmo
Requisito funcional	O software deve ter uma lista de comandos que permita ao utilizador controlar todas as operações do software
Requisito técnico	O software deverá processar os artigos o mais rápido possível e garantir que não está a processar um artigo já armazenado evitando assim blocos duplicado

Requisito funcional	O software irá filtrar todos os artigos e apenas armazenar aqueles que estão escritos em inglês
Requisito técnico	O software deverá ser capaz de lidar com os erros e exceções graciosamente e ser capaz de escrever mensagens de erros claras para o utilizador sem interromper a sua execução
Requisito técnico	O programa deverá armazenar o conteúdo da blockchain e da base de dados de classificações de artigos numa base de dados não relacional (MongoDB)
Requisito técnico	O programa deverá ser capaz de ler e escrever o conteúdo necessário para o funcionamento esperado do programa no MongoDB
Requisito técnico	O projeto deverá ser capaz de normalizar o texto dos artigos que vai ser inserido na Blockchain e o conteúdo dos artigos que o utilizador pretende comparar com o conteúdo da Blockchain.
Requisito funcional	O programa deverá conseguir dividir a totalidade dos artigos em 2 conjuntos, 30% para testes e os restantes 70% para treino para demonstrar a fiabilidade dos algoritmos implementados
Requisito funcional	O programa deverá ser capaz de exibir ao utilizador o resultado da classificação de artigos utilizando gráficos interativos
Requisito técnico	O programa deverá ser capaz de guardar os resultados das classificações de artigos em ficheiros .json
Requisito funcional	O programa deverá ser capaz de pedir ao utilizador para classificar os artigos existentes numa de 10 categorias e guardar os resultados numa base de dados própria
Requisito funcional	O programa deverá ser capaz de carregar uma base de dados já previamente categorizada com pelo menos 1000 artigos e mostrar os respetivos gráficos interativos e ficheiros .json para o utilizador analisar a fiabilidade dos algoritmos de distância métrica
Requisito funcional	O programa deverá ser capaz de fazer cópias de segurança da base de dados de classificações de artigos e da blockchain a pedido do utilizador
Requisito funcional	O programa deverá ser capaz de restaurar para o sistema as bases de dados de artigos e respetivas classificações (sempre que as mesmas existam na pasta do projeto)

A lista abaixo mostra os requisitos que foram inicialmente pensados a serem incluídos no projeto mas que devido à complexidade dos mesmos não foi possível a sua implementação.

Categoria	Descrição
Requisito funcional	O programa deverá conseguir extrair URL's de vários sites diferentes sem trazer texto desnecessário
Requisito funcional	O programa deverá conseguir extrair apenas o corpo do artigo de qualquer URL que o utilizador insira

Capítulo 4.2 Cenários de aplicação

Neste capítulo, apresentaremos alguns cenários de aplicação para o projeto de combate às notícias falsas utilizando a blockchain e algoritmos de distância métrica. Irá ser explorado como a solução desenvolvida pode ser aplicada em diferentes contextos, para que os cenários propostos sejam possíveis, era necessário que a blockchain tivesse todos os artigos mais recentes guardados de modo a poder ajudar o utilizador a verificar se uma notícia é falsa ou não, os cenários propostos são os seguintes:

Cenário de Aplicação para Redes Sociais

Um dos cenários de aplicação mais relevantes é o uso da solução em plataformas de redes sociais. Atualmente, as redes sociais são uma das principais fontes de informações para muitas pessoas, mas também são conhecidas por serem propensas à disseminação de notícias falsas. Com a integração da tecnologia blockchain e dos algoritmos de distância métrica, os utilizadores poderão verificar a autenticidade das notícias compartilhadas nestas redes.

Não é difícil de imaginar um cenário em que um utilizador encontra uma notícia alarmante ou polêmica na sua linha do temporal de uma rede social. Ele pode utilizar este projeto para verificar a veracidade dessa notícia. O conteúdo do artigo é extraído da página web e comparado com os artigos armazenados na blockchain. Os algoritmos de distância métrica comparam o artigo que o utilizador leu e ajuda o utilizador a perceber se o artigo é verdadeiro ou falso Com essa informação, o utilizador pode tomar uma decisão informada.

Verificação de Notícias nas Plataformas de Comunicação

Outro cenário de aplicação relevante é a integração da solução nas plataformas de comunicação, como aplicativos de mensagens e e-mails. As notícias falsas podem ser disseminadas rapidamente através desses dessas aplicações, e muitas vezes as pessoas partilham informações sem verificar sua autenticidade.

Ao incorporar o projeto nessas plataformas, os utilizadores podem verificar a veracidade de uma notícia antes de compartilhá-la. Por exemplo, se um utilizador receber um link para um artigo através de uma aplicação de mensagens, ele pode usar a funcionalidade do projeto para verificar se a notícia é verdadeira ou falsa. Isso ajudaria a evitar a disseminação de informações incorretas e potencialmente prejudiciais.

Aplicações em Agências de Notícias e Jornalismo

Além de ser útil para o utilizador comum, este projeto pode ser também pode ser implementado em agências de notícias e redações jornalísticas. Com a crescente preocupação com a propagação de notícias falsas, as agências de notícias podem utilizar o projeto para verificar a autenticidade das informações antes de publicá-las ao comparar o conteúdo do seu artigo com o de outras fontes jornalísticas, claro que neste cenário se a aplicação não devolver nenhum resultado isso pode querer dizer que eles vão ser os primeiros a publicar a dita noticia, o que por si também oferece uma vantagem competitiva.

Os jornalistas podem usar a ferramenta para verificar a fonte e a veracidade das notícias armazenadas. Além disso, o sistema de classificação dos artigos pode ser útil para avaliar a eficácia de diferentes algoritmos na verificação de informações em tempo real. Isso pode ajudar a melhorar os processos de verificação de fatos nos canais noticiários e jornais e garantir que apenas informações confiáveis sejam divulgadas ao público.

Uso em Plataformas de Educação e Pesquisa

A solução também pode ser aplicada em plataformas de educação e pesquisa. À medida que estudantes e pesquisadores acedem a uma ampla gama de informações online, é essencial que eles possam verificar a veracidade das fontes para evitar a utilização de dados incorretos nos seus trabalhos acadêmicos.

Com a integração do projeto em plataformas de aprendizagem e pesquisa, os estudantes podem verificar a autenticidade das informações e ter mais confiança na qualidade e precisão dos dados

utilizados nos seus estudos. Isso pode contribuir para a promoção de uma cultura de pesquisa e aprendizagem mais rigorosa e fundamentada em informações confiáveis.

Implementação nos Órgãos Governamentais

Por fim, órgãos governamentais também podem se beneficiar deste projeto para combater a propagação de notícias falsas relacionadas a políticas públicas, eleições e questões sociais. Ao adotar o projeto, os governos podem-se auxiliar deste projeto para ajudar a verificar a autenticidade das informações que circulam nas plataformas sociais e noutras outras plataformas de comunicação antes de tomar decisões baseadas em dados incorretos.

A aplicação do projeto nos órgãos governamentais pode aumentar a transparência e a confiabilidade das informações divulgadas pelos governos e, ao mesmo tempo, proteger os cidadãos de serem influenciados por notícias falsas que possam afetar negativamente a sociedade.

Conclusão

Os cenários de aplicação apresentados demonstram o potencial do projeto de combate às notícias falsas com o uso de blockchain e algoritmos de distância métrica. Esta solução pode ser implementada em várias áreas para ajudar a verificar a veracidade das informações e promover a disseminação de conteúdo autêntico e confiável. A integração dessa tecnologia em plataformas de redes sociais, comunicação, jornalismo, educação, pesquisa e governos pode ajudar a combater a propagação de notícias falsas e contribuir para um ambiente de informações mais transparente e confiável.

Capítulo 4.3 Diagramas

Nesta secção encontram-se os diagramas referentes às principais funcionalidades do projeto.

O primeiro diagrama mostra que elementos dos artigos são ou não considerados quando os mesmos são extraídos. Caso o corpo do artigo não esteja em inglês o mesmo é descartado automaticamente, senão é adicionado à base de dados blockchain.

Ao limitar os artigos guardados e aceite para analise pelo programa para apenas inglês, ajudará bastante à sua performance tendo em conta que a maioria dos projetos desenvolvidos são pensados para texto só em inglês e outras línguas podem eventualmente causar um grande número de falsos negativos ou positivos indesejados.

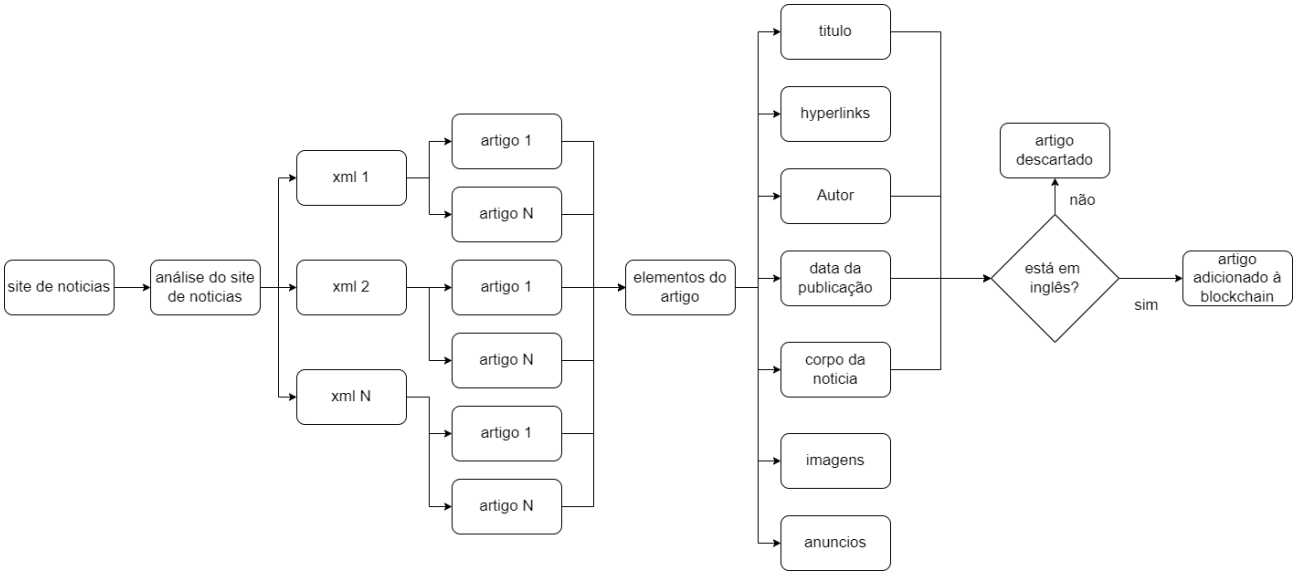


Figura 17 - Extração de artigos

O segundo grupo de funcionalidades refere-se à construção e inserção dos dados dentro da blockchain. O primeiro passo consiste na criação da blockchain e do bloco genesis, sendo este um bloco especial porque não irá conter dados e não terá nenhum bloco antes dele.

Após a blockchain ser inicializada os blocos poderão começar a ser criados e cada bloco terá dentro dele o URL do artigo, o seu título, a sua data de publicação, o seu autor, o corpo do artigo legível para pessoas normais e o corpo do artigo normalizado também será inserido na base de dados juntamente com a hash do bloco que lhe antecede, formando assim uma ligação entre as informações de cada artigo que formam um bloco.

Ao mesmo tempo será gerada a hash do deste bloco de modo a esta ser inserida no próximo bloco gerando assim o encadeamento entre eles.

A blockchain por sua vez também permite que seja verificado se o conteúdo de nenhuma hash foi alterada garantido assim a integridade da mesma.

O processo descrito anteriormente pode ser observado na imagem abaixo:

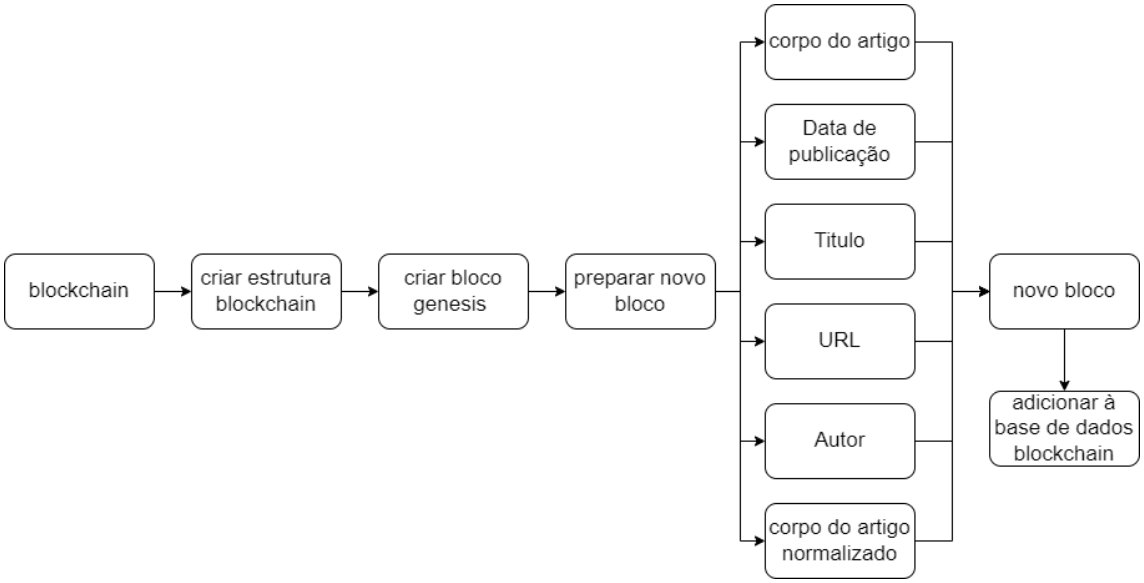


Figura 18 - Criação dos blocos

Outra funcionalidade já anteriormente mencionada é a possibilidade de comparar um artigo do utilizador com o conteúdo da blockchain utilizando os algoritmos de distâncias métrica implementados e indicar ao utilizador qual o artigo mais semelhante encontrado.

A imagem abaixo mostra uma representação de como o código analisa o pedido do utilizador e devolve o resultado.

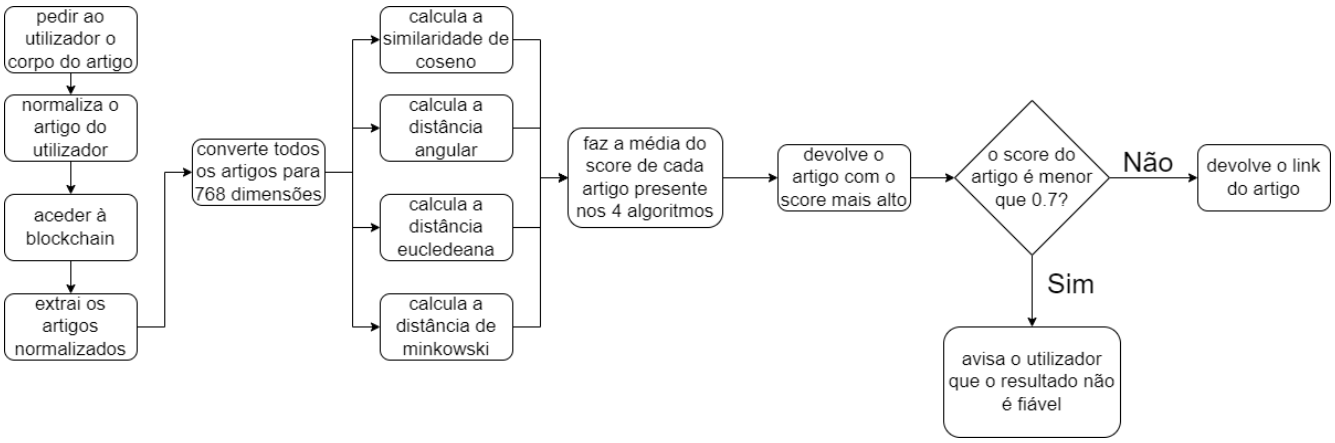


Figura 19 - Verificar veracidade dos artigos

Por fim falta a funcionalidade de classificar os artigos manualmente e ver se os algoritmos de distância métrica conseguem ou não calcular com sucesso uma nova categoria para o todos os artigos do conjunto dos artigos de teste (30%). Cada artigo é considerado corretamente classificado quando a categoria calculada pelo programa é igual à categoria que o utilizador indicou. O esquema abaixo representa simplificadamente o processo que o programa faz para exibir as categorias calculadas.

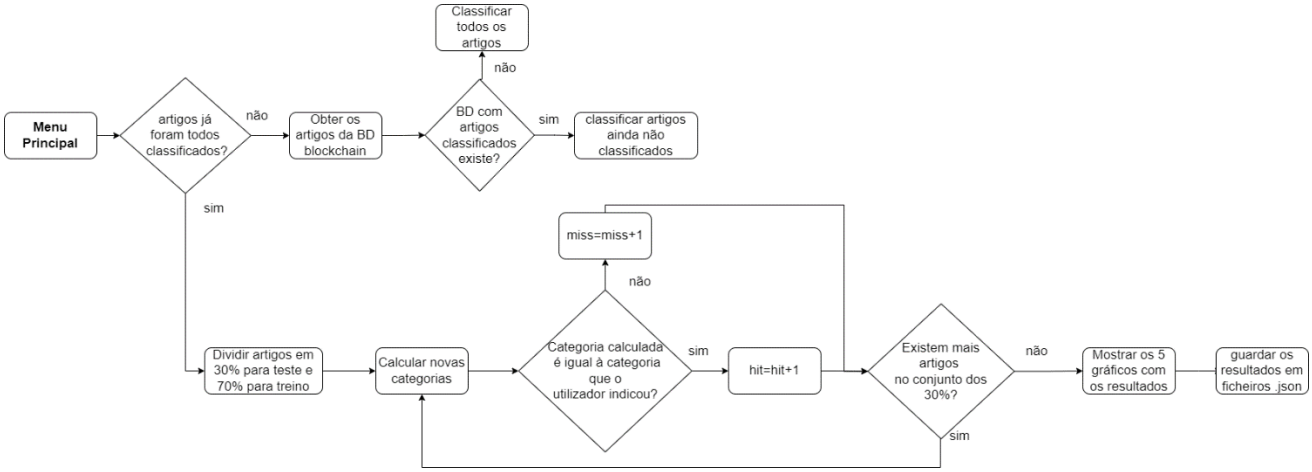


Figura 20 - Cálculo das novas categorias

O processo para a base de dados já com os 1137 artigos classificados segue exatamente o mesmo caminho representado no diagrama acima mas passa imediatamente para o cálculo das categorias uma vez que todos os artigos já estão classificados.

Capítulo 5. Solução proposta

Neste capítulo serão abordadas serão analisadas as principais funcionalidades do programa com recurso a diagramas. Ainda neste capítulo serão abordados os pedaços de código mais relevantes do projeto e a sua respetiva taxa de esforço.

Será também abordado a tecnologia utilizada no projeto e o porquê de terem sido escolhidas e a respetiva fundamentação das principais opções tecnológicas. Também será feito um enquadramento da arquitetura do projeto.

Todo o projeto estará disponível no repositório (https://github.com/FilipeCacho/Fake-News-Blockchain) e poderá ser corrido num IDE que tenha instalado o Python 3.8. Todas as instruções necessárias de como instalar e correr o projeto podem ser encontradas no repositório indicado acima, assim como uma descrição geral do trabalho e dos seus objetivos e funcionalidades. O vídeo com uma pequena demonstração do projeto pode ser acedido utilizando este link: https://youtu.be/VjXWjTkwpS8

Capítulo 5.1. Arquitetura

Neste capítulo vai ser descrito de modo geral como é que os principais componentes do projeto se relacionam, a lógica do fluxo de dados e as interações entre os diversos módulos do projeto.

Na sua base este projeto consiste na extração de artigos para estes serem colocados numa base de dados com a estrutura de uma blockchain, para depois eles serem devidamente classificados para se poder testar os algoritmos de distância métrica implementados, e portanto todos os módulos interagem sem exceção com as bases de dados criadas.

O esquema abaixo mostra de modo geral e simplificado a dependência de todos os módulos MongoDB que é o sistema de base de dados que permite a criação da base de dados blockchain e da base de dados das classificações.

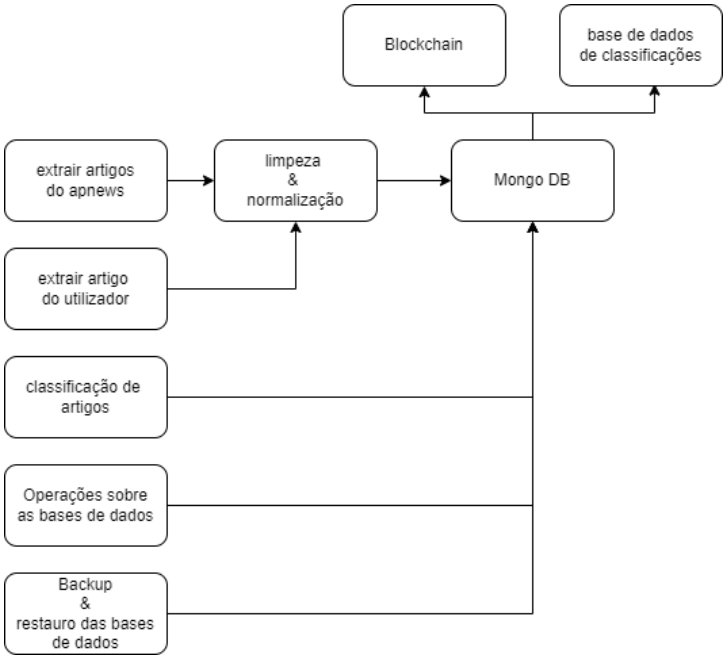


Figura 21 - Dependência dos módulos do mongoDB

No que toca ao fluxo de dados dentro do projeto, não existe uma interação recursiva ou constante entre os diversos módulos do projeto sendo a única exceção a inserção de blocos na blockchain. Apesar de existir comunicação e passagem de dados entre os diversos módulos do projeto, isso acontece apenas uma única vez e depende de que opção do menu é invocada pelo utilizador, só no caso indicado abaixo é que existe uma passagem recursiva dos dados entre diferentes módulos do programa.

O esquema abaixo mostra a passagem de dados entre os diferentes módulos da aplicação quando está a ser realizado o processo de extração de artigos do website apnews.com

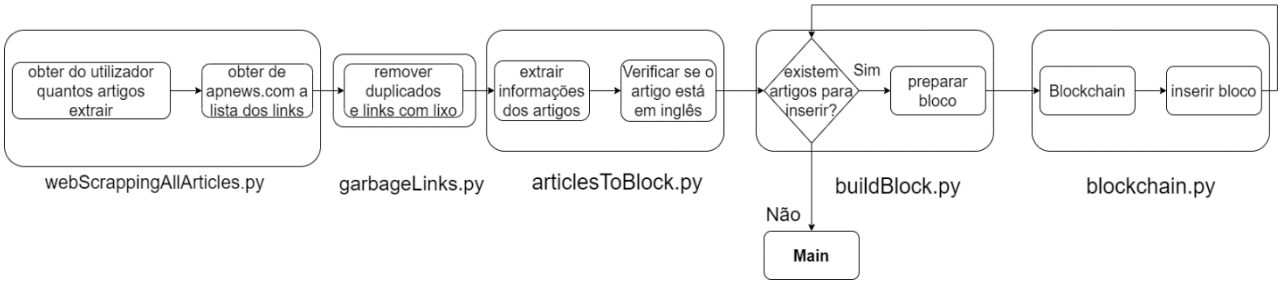


Figura 22 - Processo de criação de blocos

No resto do programa a passagem de dados acontece entre diferentes funções dentro do mesmo módulo, isto porque sendo operações especificas daquele modulo que não são usadas em mais lado nenhum não existiu necessidade de criar módulos separados para colocar essas funções.

Claro que não podemos esquecer que todos os módulos interagem com o sistema externo aonde estão as bases de dados que neste caso é o mongoDB como pode ser visto na figura 21.

As funcionalidades da aplicação dividem-se sobretudo em 3 grupos, sendo estes a extração e processamento de artigos, a criação da blockchain e devido armazenamento da mesma no MongoDB e todas as operações de interação com a base de dados.

As funcionalidades encontram-se mais detalhadas abaixo:

- **Processamento dos artigos**, consiste em todas as funcionalidades relacionadas com a inspeção e extração dos artigos selecionados. Existem inúmeros tarefas a serem realizadas durante este processo, como a análise de todo o mapa do site, a exclusão de páginas que não contem artigos ou páginas com conteúdo descartável, a extração e formatação das componentes dos artigos consideradas necessárias, para depois ser verificado se estão em inglês e se sim serão inseridas num bloco que irá ser cifrado para depois ser inserido na Blockchain.

- **Conexão com o MongoDB,** consiste em todas as funcionalidades que validam se existe conexão à BD aonde se encontra a Blockchain. Ao longo do programa é verificado se o serviço MongoDB está ativo e se sim se a base de dados blockchain já existe. Isto é necessário porque o projeto tem de estar preparado para o caso da base de dados blockchain ainda não existir, o que leva o programa a assumir que a mesma ainda não foi criada e, portanto, todos os artigos terão de ser inseridos outra vez, o que faz com que o programa salte por exemplo a verificação de artigos duplicados que já possam existir na base de dados Blockchain.

- Blockchain, consiste num único módulo que é chamado após ter sido extraído as informações de um artigo (título, URL, autor, etc.) para estas serem agregadas num bloco para ser adicionado à blockchain BD. Neste módulo é gerado a prova de trabalho (proof-of-work) e a cifra do bloco que o antecede. É também neste bloco que é inicializado o bloco genesis, cujo índice é 0, este bloco é necessário porque o primeiro artigo a ser inserido tem de conter a cifra de algum bloco, sendo essa a funcionalidade do bloco genesis.

Os restantes blocos são adicionados com um índice igual ao índice do bloco que o antecede +1 e contêm a respetiva hash do bloco que lhe antecede, garantido assim a

correta inserção de todos os blocos seguintes.

- **Comparar artigo do utilizador,** é aqui que uma das mais importantes funcionalidades do programa é utilizada, que serão os algoritmos de distância métrica. O programa permite ao utilizador inserir os vários parágrafos que constituem o corpo de uma notícia. Todos esses parágrafos são normalizados para facilitar a análise pelos algoritmos de distância métrica.

Após isso são chamados os 4 algoritmos de distância métrica implementados, todos capazes de fazer diferentes tipos de análises sobre o texto do utilizador. Simultaneamente estes módulos vão buscar os corpos normalizados de todos os artigos armazenados na BD um a um e aplicam os algoritmos de processamento de distância métrica. Cada um dos 4 algoritmos devolve uma lista com os 5 artigos mais semelhantes em relação ao artigo do utilizador, depois é escolhido o artigo que se encontra presente na lista de todos os 4 algoritmos. Caso exista mais do que um artigo que se repete na lista dos 4 algoritmos de distância métrica, é calculado a média da soma do score desse artigo, o resultado devolvido é o artigo com o score mais alto. Caso o score final seja abaixo de 0.6, o programa devolve o resultado na mesma mas avisa que o resultado não é alto o suficiente para se ter a certeza de que o artigo devolvido é o correto, apenas que é o artigo mais semelhante encontrado.

Quanto mais próximo o resultado está ao valor de 1, mais semelhante deverá ser o artigo do utilizador em relação ao artigo devolvido pela blockchain.

- **Exportar as bases de dados para .json,** é uma funcionalidade desenvolvida apenas para exibir toda a blockchain num formato mais amigo do utilizador. Consiste em ler ambas as bases de dados e exportá-las para um ficheiro JSON localizado na pasta do projeto. A funcionalidade não serve só para observar resultados como também serve para mostrar ao utilizador que não temos nada a esconder e que o conteúdo da base de dados é publica.

- **Apagar a base de dados blockchain ou a base de dados das classificações,** serve para testar várias funcionalidades do programa, incluindo ver como o mesmo se comporta quando a base de dados não existe, também permite ver o comportamento do programa quando ele tem que baixar todos os artigos ou se salta os artigos duplicados.

- Apagar um bloco da blockchain, é uma funcionalidade desenvolvida para permitir observar se a funcionalidade de verificar se a base de dados blockchain foi indevidamente modificada ou não uma vez, que apagando um bloco isto vai fazer com que a verificação das hashes falhe.

- Fazer cópias de segurança e restaurar as bases de dados, são funcionalidades que permitem ao utilizador testar o programa sem ter de fazer o download de todos os artigos cada vez que o programa arranca. No futuro esta funcionalidade poderá ser facilmente adaptada para permitir restaurar a base de dados automaticamente caso seja detetada alguma inconsistência nalgum bloco da blockchain.

- Dividir os dados em 30% para teste e 70% para treino, permite ao utilizador verificar a eficácia dos algoritmos de distância métrica ao apresentar os gráficos e ficheiros aonde se consegue ver as categorias que o programa calculou automaticamente juntamente com a percentagem de acertos e erros.

Capítulo 5.2. Tecnologias, ferramentas utilizadas e fundamentação

A linguagem de escolha para o desenvolvimento do projeto foi o Python por oferecer acesso a um conjunto de bibliotecas de processamento de texto e por ser a linguagem de programação de referência para algoritmos de distância métrica. Como foi notado nas aulas de inteligência artificial o Python é uma linguagem extremamente acessível, mas bastante poderosa e é das melhores (senão a melhor) linguagem para lidar com a interpretação e analise de grandes quantidades de dados.

Das inúmeras funcionalidades disponíveis pelo Python, até agora a biblioteca mais importante deste projeto disponibilizada pela comunidade é o "**BeautifulSoup**", que é utilizada para recolher todos os artigos dos websites e fazer a processamento dos mesmos de modo a extrair a apenas o conteúdo necessário. É uma biblioteca bastante acessível que permite invocar os elementos html das páginas web a partir do seu nome, permitindo assim recolher exatamente os elementos que eu preciso das páginas web com facilidade.

Outras bibliotecas utilizadas que são uma parte integral deste projeto são:

- Langdetect, uma biblioteca que permite inserir um texto e que devolve a linguagem em que o texto está, no meu caso é usado para excluir artigos não em inglês.

- **Hashlib**, é uma biblioteca que permite fazer hashes, é usada no projeto para criar as hashes SHA-256 para a blockchain

- **Json**, é a biblioteca usada que permite ler e escrever os ficheiros .json aonde são guardados os resultados do programa.

- **Pymongo**, é a biblioteca que permite fazer a conexão com o mongoDB instalado no sistema, é indispensável para o funcionamento do projeto.

- **Scipy.spatial.distance**, implementa a similaridade do cosseno, distancia euclidiana e distancia de Minkowski.

- **Nltk.corpus**, ajuda a remover palavras stop do texto o que simplifica a sua composição gramatical, o que aumenta o desempenho dos algoritmos de distância métrica.

- **Nltk.stem**, remove acentos e símbolos especiais e reduzir as palavras à sua forma mais básica o que também contribui para um melhor desempenho dos algoritmos de distâncias métricas.

- **Matplotlib**, é a biblioteca utilizada para desenhar os 5 gráficos interativos do programa.

- **SBERT,** é a biblioteca que reduz o corpo dos artigos para um número específico de dimensões o que aumenta bastante o desempenho dos algoritmos de distância métrica implementados ao transformar o texto num conjunto de vetores com dimensão fixa mas aonde o contexto principal do artigo não é perdido.

A blockchain foi criada com recurso a vários exemplos encontrados da internet, uma vez que se trata de uma estrutura de dados que pode ser adaptada consoante a situação, foi necessário analisar vários exemplos para se perceber o seu funcionamento, para se poder criar uma blockchain adaptada ao projeto.

Para programar o projeto foi utilizado o Windows 10 e foi escolhido o Visual Studio Code como editor de código. Os critérios de escolha foram simples, tinha de ser gratuito, suportar Python e ambientes virtuais. O suporte a ambientes virtuais é bastante importante porque vai permitir que o projeto seja corrido em qualquer computador apenas tendo de ser instalado a versão do Python indicada, o mongoDB no sistema e os requisitos indicados no repositório Git.

Antes de se puder começar a comparar os artigos entre si ou com o artigo inserido pelo utilizador é necessário processar todo o texto de modo a remover palavras redundantes e

a prepará-lo para analise.

Para tal, como já foi dito anteriormente são removidas as palavras stop, também são removidos todos os caracteres especiais, mas apos isso é preciso reduzir o texto a um número específico de dimensões para que este fique na forma de um vetor para poder ser analisado e encontradas as semelhanças entre 1 ou mais textos, para tal é usado o SBERT neste projeto.

O SBERT (Sentence-BERT) é um modelo de codificação de frases pré-treinado que utiliza a arquitetura do BERT (Bidirectional Encoder Representations from Transformers) para gerar representações de alta qualidade para frases.

O SBERT consegue a capturar a semântica das frases, considerando o contexto e as relações entre as palavras. Utiliza uma abordagem de codificação bidirecional, o que significa que leva em consideração tanto o contexto anterior quanto o posterior de cada palavra na frase. No projeto todos os artigos comparados são reduzidos a 768 dimensões antes de serem passados para os algoritmos de calculo de distâncias métricas.

Toda a comparação dos artigos do utilizador com os guardados na blockchain é feita com recurso a algoritmos de processamento de distância métrica. Existem vários algoritmos possíveis, mas, no entanto, os algoritmos escolhidos para comparar se o texto escolhido pelo utilizador é semelhante ou contraditório com algum artigo da BD são os seguintes:

Similaridade de Cosseno:

A similaridade de cosseno é uma medida que calcula a proximidade entre dois vetores, independentemente do seu tamanho. Mede o cosseno do ângulo entre os vetores, variando de -1 (completamente opostos) a 1 (totalmente semelhantes). Quanto mais próximo o valor for de 1, maior a similaridade entre os vetores e portanto maior é a similaridade entre 2 artigos.

Distância Angular:

A distância angular é uma medida que calcula a diferença angular entre dois vetores em relação a um ponto de referência. Ela é calculada a partir do arco cosseno da similaridade de cosseno, normalizado para um intervalo de 0 a 1. Quanto menor o valor da distância angular, maior a similaridade entre os vetores.

Distância Euclidiana:

A distância euclidiana é uma medida que calcula a distância direta entre dois pontos em um espaço euclidiano. Ela é calculada como a raiz quadrada da soma dos quadrados das diferenças entre as coordenadas dos pontos. Quanto menor a distância euclidiana, mais próximos os pontos estão no espaço.

Distância de Minkowski:

A distância de Minkowski é calculada elevando cada diferença de coordenada à ordem especificada, somando esses valores e, em seguida, tirando a raiz dessa soma.

Outra componente importante deste projeto é a Blockchain e os seus algoritmos de decisão, mas ao mesmo tempo é uma solução tecnológica e, portanto, as razões que levaram à sua escolha serão abordadas neste capítulo e não no anterior.

Começando pela blockchain, que essencialmente não é nada mais nada menos do que lista que mantem um conjunto de registos que estão sempre a ser atualizados, a esses registos dá-se o nome de blocos (blocks). Cada bloco está ligado ao bloco que lhe antecede através de uma cifra criptográfica. Para além disso cada bloco deverá conter os dados da transação e a data e hora da dita transação. A única exceção a esta regra é o bloco zero (o primeiro bloco da Blockchain) que não contem a cifra do bloco anterior. [15]

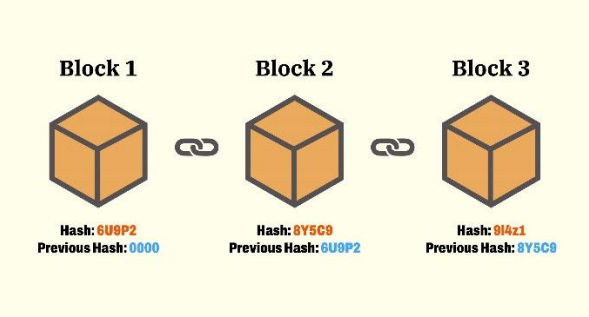


Figura 23 - Representação da Blockchain [16]

A funcionalidade mais interessante da Blockchain é o facto de cada bloco conter uma cifra do bloco anterior. Esta cifra é gerada encriptando o conteúdo todo do bloco anterior e qualquer tentativa de modificar o bloco anterior gera uma cifra completamente diferente e, portanto, essa transação é recusada. Este mecanismo também não permite que outros blocos sejam inseridos à força. Tornando a Blockchain na teoria inviolável.

Na prática as Blockchain são mantidas por centenas ou milhares de participantes (também conhecidos por “nodes”) que processam as transações e todos eles tem uma lista de todos os blocos e as respetivas cifras.

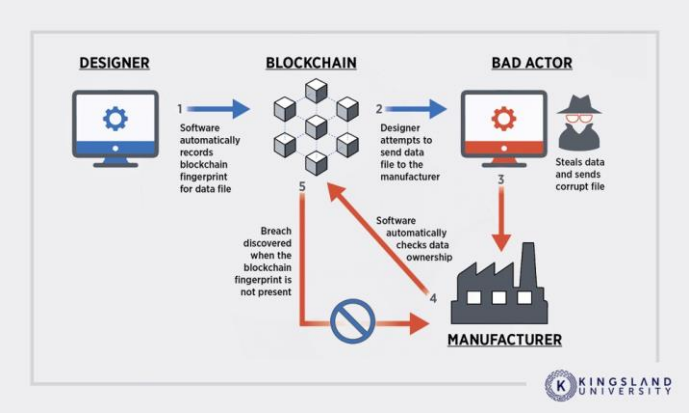


Figura 24 - Exemplo de uso da Blockchain [17]

Para alguém modificar um único bloco, teria de ser ou tomar controlo de pelo menos 51% dos nodes participantes para falsificar os registos. Como na maioria dos casos os participantes são descentralizados e espelhados pelo mundo isso torna praticamente impossível falsificar os blocos pelo menos utilizando métodos convencionais. [18]

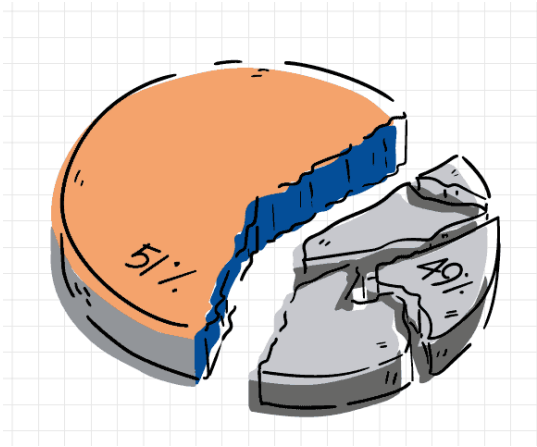


Figura 25 - Representação da maioria dos nodes (52%) [18]

É de notar que apesar de todas as vantagens que uma Blockchain tem existem vários possíveis ataques que podem ser realizados com o intuito de falsificar dados [19].

Mas para o âmbito do projeto e uma vez que se trata de um repositório de notícias e não Blockchain para transação de cripto moedas (que são as mais atacadas) que as desvantagens são superadas pelas vantagens, permitindo assim um sistema robusto,

rápido, de fácil inserção e consulta de dados e que foi construído tendo em mente que os dados contidos seriam fidedignos.

Para programar e manter a Blockchain será utilizado a linguagem de programação Python através da utilização do Visual Studio. A escolha do Python foi devido ao facto de que se trata de uma linguagem de programação já com bastantes framework construídas prontas para analisar páginas web e capazes de extrair o conteúdo pretendido (título e corpo da notícia). Também é uma linguagem capaz de suportar a criação de Blockchain como descrito no ponto abaixo, tornando-a a linguagem ideal para este projeto.

Capítulo 5.3. Implementação

Neste capítulo serão analisadas as principais opções técnicas do programa.

Começamos com a funcionalidade de extrair artigos do website. No início do projeto foi pensado que seria fácil de se extrair uma grande quantidade ou todos os artigos de vários websites diferentes, mas essa ideia foi rapidamente abandonada quando foi analisada a estrutura html de 5 páginas web de diferentes sites de notícias e rapidamente ficou obvio que estes sites guardam as notícias de forma totalmente diferente. Foi preciso, portanto, construir um modulo diferente para extrair a informação de cada artigo. A maioria dos websites de notícias contêm um ficheiro com um nome parecido a "sitemap.xml" aonde guarda os links das últimas noticias que quer que os programas de buscas usem. Portanto a primeira etapa foi extrair todos os links desses websites.

O excerto de código abaixo permite a extração de todos os URL's das notícias mais recentes que o site https://apnews.com quer disponibilizar. O código lê o ficheiro .xml e extrai todos os links que se encontram dentro da tag html "<loc>"

```python
def webScrapingAllArticles(max_links=0):
    all_urls = []
    max_links_reached = False

    num_urls = 0  # Add a counter for the number of URLs found

    for sitemap_url in sitemap_urls:
        response = requests.get(sitemap_url)
        soup = BeautifulSoup(response.content, "xml")

        if soup.find("urlset") is not None:
            loc_tags = soup.find_all("loc")
            urls = [loc.get_text() for loc in loc_tags]
            num_urls += len(urls)  # Increment the counter
            time.sleep(1)

            for j, url in enumerate(urls):
                time.sleep(0.002)

                if max_links > 0 and len(all_urls) >= max_links:
                    max_links_reached = True
                    break

                # Add the URL to the list
                all_urls.append(url)

                # Print the index of the processed URL
                print(f"  Processing URL {len(all_urls)}/{num_urls}: {url}")

            if max_links_reached:
                break
```

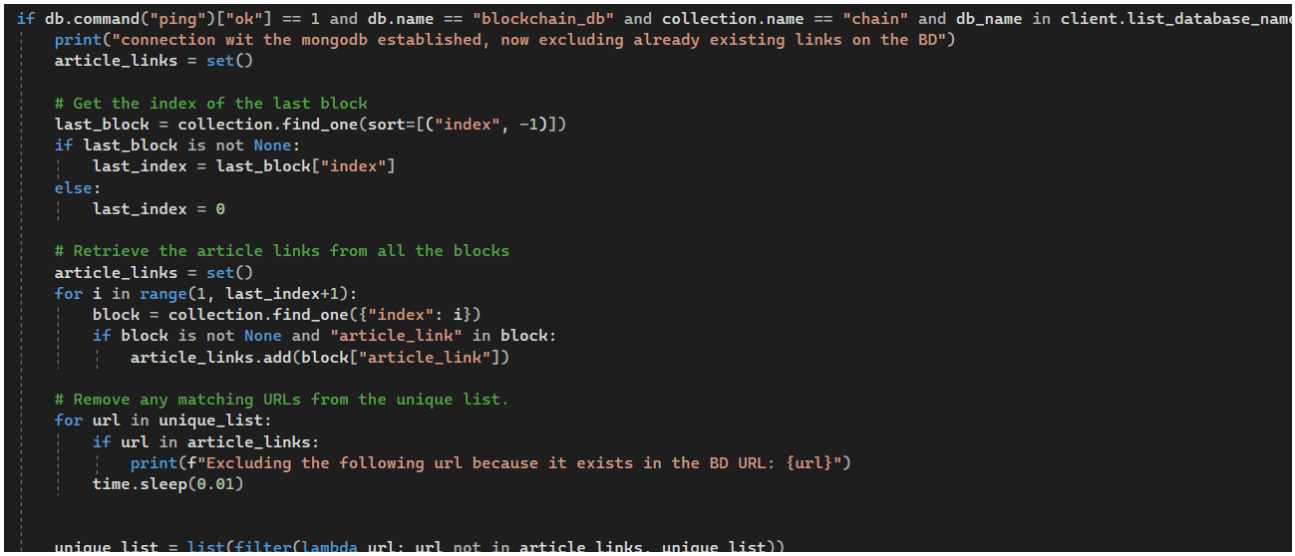
Figura 26 - extrair URLS

De seguida os links extraídos passam por um modulo que retira os links que contem certas palavras, isto porque existem centenas de links que não contem artigos e, portanto, são inúteis, e depois disso a lista de links é passada pela estrutura de dados dicionário para remover valores duplicados.

```python
def garbageLinks(unique_list):
    os.system('cls')

    print(len(unique_list))
    print("tamanho da lista de links com lixo")
    time.sleep(1.7)

    garbage = {'https://apnews.com/hub/'}

    cleanedList = list(filter(lambda url: not any(ignore in url for ignore in garbage), unique_list))

    # Remove duplicated links by passing them through a dict. Dicts in Python <3.7 preserve order.
    unique_dict = OrderedDict.fromkeys(cleanedList)

    unique_list = list(unique_dict.keys())
```

Figura 27 - Remover lixo

Ainda dentro do modulo que remove o lixo, é realizada a conexão à base de dados blockchain para comparar os URL's já existentes com os URLS que foram acabados de extrair do "sitemap", caso já existam na base de dados estes são removidos da lista de links a serem extraídos



```
if db.command("ping")["ok"] == 1 and db.name == "blockchain_db" and collection.name == "chain" and db_name in client.list_database_name
    print("connection wit the mongodb established, now excluding already existing links on the BD")
    article_links = set()

    # Get the index of the last block
    last_block = collection.find_one(sort=[("index", -1)])
    if last_block is not None:
        last_index = last_block["index"]
    else:
        last_index = 0

    # Retrieve the article links from all the blocks
    article_links = set()
    for i in range(1, last_index+1):
        block = collection.find_one({"index": i})
        if block is not None and "article_link" in block:
            article_links.add(block["article_link"])

    # Remove any matching URLs from the unique list.
    for url in unique_list:
        if url in article_links:
            print(f"Excluding the following url because it exists in the BD URL: {url}")
        time.sleep(0.01)


    unique_list = list(filter(lambda url: url not in article_links, unique_list))
```

Figura 28 - Remover duplicados

Após todos os ser construída uma lista sem links em duplicados nem com lixo para serem extraídos, esta lista vai entrar para o modulo que extrai a data, o URL, o título, etc. Como já foi dito anteriormente é necessário um modulo para extrair noticias de um website específico, uma vez que cada website guarda as notícias de forma diferente no html. O código apresentado abaixo mostra como é feita a extração dos artigos do site “Apnews” e só funciona para este site.

Para tal é necessário analisar o código de vários artigos deste site e tentar encontrar a semelhança entre os mesmos. Este é um aspeto fundamental do projeto e foi preciso afinar este modulo através de tentativa e erro ao analisar o resultado obtido várias vezes até se obter apenas o conteúdo pretendido sem trazer lixo.

```python
def process_url(url):
    try:
        html_content = requests.get(url).text

        soup = BeautifulSoup(html_content, "html.parser")

        # Find the title tag.
        h1_tag = soup.find('h1', class_=lambda x: x and 'component-heading' in x.lower())
        h2_tag = soup.find('h2', class_=lambda x: x and 'component-heading' in x.lower())

        if h1_tag:
            title = h1_tag.text
        elif h2_tag:
            title = h2_tag.text
        else:
            title = "Title not found"

        normalized_title = unidecode(title)
        cleaned_title = re.sub(r"[^a-zA-Z0-9,.?!;:()\'\"/\\\-_ \n\t]", "", normalized_title)
        cleaned_title = re.sub(r"\n{2,}", " ", cleaned_title)

        # Find the publication date.
        time_element = soup.select('span[class*="Timestamp Component"]')
        date = str(time_element[0])
        parts = date.split(">", 1)
        remaining_text = parts[1]
        remaining_text_parts = remaining_text.split("</span", 1)
        publication_date = remaining_text_parts[0]

        # Find the author.
        bylines = soup.find_all('span', {'class': lambda c: c and 'Component-bylines' in c})
        if bylines:
            author = re.search(r'>([^<]+)<', str(bylines))
            author_name = author.group(1)
        else:
            author_name = "no author"
```

Figura 29 - Extrair partes do artigo

Outro aspeto importante que se encontra neste modulo é a funcionalidade de detetar se o conteúdo do artigo que vai ser adicionado à blockchain se encontra em inglês ou não. Esta funcionalidade é possível graças à biblioteca “langdetct” que é alimentada com o corpo de cada artigo antes de este ser adicionado à blockchain e se existir uma conexão à net é capaz de devolver a linguagem do texto submetido. Caso o artigo esteja em inglês então este pode ser adicionado à blockchain.

```python
try:
    if str(blockchain_info[-1])!="No article text found" or len(str(blockchain_info[-1])) == 0 or str(blockchain_info[-1]).issp
        language = detect(blockchain_info[-1])

        if  language == 'en' and garbageTitles(str(blockchain_info[0]))==False:
            buildBlock(blockchain_info)
            print("Block added to the chain")
            blockchain_info.clear()

        elif language!='en':
            print("Block not added because its not in english")
            blockchain_info.clear()
```

Figura 30 - Detetar linguagem do corpo do artigo

De seguida cada site que passa todas as validações tem o corpo do artigo normalizado, ou seja, é removido a pontuação, palavras stop e o texto é reduzido à sua forma essencial e

básica, isto ajuda os algoritmos de processamento de distância métrica a conseguirem melhores resultados uma vez que o lixo sintático e gramático é removido.

Após isto todo o conteúdo extraído de cada página web é passado no modulo da blockchain que gera cada bloco com as respetivas informações.

Na imagem abaixo é mostrado parte do código da blockchain que gera a prova de trabalho, ou seja a cifra que serve de prova que o bloco passou por um desafio criptográfico para ser gerado. Para este projeto e devido ao hardware limitado, a prova de trabalho foi criada para ser simples, porque senão iria levar demasiado tempo a calcular a prova para cada artigo. Caso alguém decida continuar o projeto e tenha hardware mais avançado pode ir ao modulo da blockchain e modificar a fórmula.

```
def _hash(self, block: dict) -> str:
    # Convert the ObjectId to string
    block["_id"] = str(block["_id"])
    encoded_block = _json.dumps(block, sort_keys=True).encode()
    return _hashlib.sha256(encoded_block).hexdigest()

def _to_digest(
    self, new_proof: int, previous_proof: int, index: str, articleTitle: str, articleDate: str, articleAuthor: str, articleLink: str, ar
    to_digest = (str(new_proof**2 - previous_proof**2 + index) + articleTitle  + str(articleDate) + articleAuthor + articleLink + articl
    return to_digest.encode()

def _proof_of_work(
    self, previous_proof: int, index: int, articleTitle: str, articleDate: int, articleAuthor: str, articleLink: str, articleBody: str,
    new_proof = 1
    check_proof = False
    while not check_proof:
        to_digest = self._to_digest(new_proof, previous_proof, index, articleTitle, articleDate, articleAuthor, articleLink, articleBody
        )
        hash_value = _hashlib.sha256(to_digest).hexdigest()
        if hash_value[:4] == "0000":
            check_proof = True
        else:
            new_proof += 1
    return new_proof
```

Figura 31 - Prova de trabalho

Como já foi descrito anteriormente, o projeto é capaz de gerar um ficheiro .json em que imprime toda a blockchain (se ela existir).

Na próxima imagem, temos um excerto do ficheiro .json aonde se consegue ver o bloco genesis e os blocos seguintes da blockchain.

Podemos ver que cada bloco é constituído pelos elementos do artigo como o título, o corpo do artigo, etc. e outros elementos necessários como a “previous hash” do bloco que lhe antecede, sendo esta a principal particularidade da blockchain.

Dentro ainda de cada bloco temos o texto normalizado do artigo, este é a versão otimizada do texto que vai ser passado e para os algoritmos de distância métrica.

```
"_id": "64aaee77453de01a66f9844a",
"index": 2,
"blockTimestamp": "2023-06-25 18:09:52.194652",
"proof": 66975,
"previous_hash": "5dbd557abaff20980a3e4d8405fe398197679ffc721d217aff1163133378d51b",
"article_title": "6clicks Expands Offering with Launch of Marketplace for GRC Vendors and Advisors",
"article_date": "January 17, 2023 GMT",
"article_author": "no author",
"article_link": "https://apnews.com/article/technology-united-kingdom-marketersmedia-australia-business-0e45c903a61cc09c0c4fd80e3d40f06a",
"article_body": "6clicks fuels growth for advisors and businesses with the launch of the first global risk and compliance marketplace for tec
--6clicks revolutionizes the Governance, Risk, and Compliance industry with the launch of 6clicks Marketplace. The platform offers businesses
content that connect with the 6clicks core GRC platform. This ground-breaking launch positions 6clicks as the go-to destination for all thing
that can help protect businesses and demonstrate compliance. The platform offers solutions for GRC issues such as cybersecurity, risk managem
support from key customers in Australia, the United States, and the United Kingdom, with over 95 of 6clicks' 1000 customers already using con
launch, there will be over a hundred partners offering access to their apps and content via the 6clicks Marketplace.Key facts about the 6clic
Head of Digital Ecosystem, Elaine Suezo, says  \"Our ecosystem is rapidly expanding with hundreds of partners eager to join. We've invested h
connect and share data with 6clicks.\" CyberCX CEO, John Pataridis, says, \"Our partnership with 6clicks has already proven successful in hel
forward to expanding our reach and helping even more businesses through this platform.\"Orpheus Cyber CEO, Oliver Church, says, \"6clicks' de
to help organizations understand and reduce their attack surface. We believe this partnership will give our products greater visibility and a
says, \"6pillars is delighted to be featured with 6clicks as a solution that delivers the continuous compliance that standards including SOC
industries requiring high levels of compliance the powerful combination to implement automated best-practice across their AWS cloud environme
implementation and ease of use. Our unique Hub  Spoke architecture allows for federated or distributed deployment, making it ideal for large
compliance and non-compliance in seconds, streamlining the process for compliance professionals. With fully integrated content, there's no ne
the name suggests (read: \"The founder's story: How 6clicks was born and what's behind the name\"), 6clicks makes it easy to manage risk and
taking on giants like ServiceNow, OneTrust, RSA Archer and Galvanize.Contact Info:  Name: Elaine Suezo  Email: Send Email Organization: 6clic
https://www.youtube.com/watch?vuX4c7L_ERwERelease ID: 89088194If you detect any issues, problems, or errors in this press release content, ki
the next 8 hours.",
"normalized_body": "0click fuel growth advisor busines launch first global risk complianc marketplac technolog servic insur content providers
marketplac platform ofer busines softwar vendor advisor manag servic provid ace wide rang ap content conect 0click core grc platform groundbr
tol resourc help protect busines demonstr complianc platform ofer solut grc isu cybersecur risk manag colabor proce optimisationth 0click mar
alreadi use content 0click librari integr ap partnership zapier launch wil hundr partner ofer ace ap content via 0click marketplacekey fact 0
suezo say ecosystem rapidli expand hundr partner eager join weve invest heavili grc platform provid custom tol manag risk complianc conect sh
improv cybersecur risk manag strategi loke forward expand reach help even busines platform orpheu cyber ceo oliv church say 0click dedic risk
believ partnership wil give product greater visibl asist busines improv cybersecur strategi 0pilar ceo lorenzo modesto say 0pilar delight fea
0click 0pilar togeth give industri requir high level complianc power combin implement autom bestpractic acro aw cloud environ u 0click cuting
distribut deploy make ideal larg enterpris advisor msp hailey ai engin use aipow mape quickli identifi complianc noncompli second streamlin p
platforma name sugest read founder stori 0click born what behind name 0click make easi manag risk complianc design advisor busines power ai i
elain suezo email send email organ 0click adr melbourn vic australia websit url id 0if detect isu problem eror pre releas content kindli cont
```


Figura 32 – Excerto do ficheiro .json da blockchain

Passamos então à classificação e exibição dos gráficos interativos, como já foi dito antes o utilizador pode escolher classificar os artigos manualmente e testar a fiabilidade dos algoritmos, ou pode ver os resultados utilizando a base de dados de artigos já previamente classificada por mim.

Os artigos podem ser classificados em 10 categorias diferentes, para tal é necessário aceder à base de dados blockchain e extrair todos os artigos para serem classificados, a imagem abaixo é um excerto do código que faz isso mesmo.

```python
# Iterate over articles in the blockchain db
for block in collection.find(no_cursor_timeout=True):
    if "article_link" in block and "article_body" in block:
        if block["article_title"] == "I'm the genesis block":
            continue  # Skip the genesis block

        # Check if the article exists in the classifications database
        existing_classification = collectionCat.find_one({'article_link': block['article_link'], 'article_title': block['article_title']})
        if existing_classification:
            totalArticles -= 1
            continue  # Skip the article if it already exists in the classifications database

        # Increment the counter for non-genesis blocks
        counter += 1

        # Get user input for the category
        os.system('cls')
        print(f"Article link: {block['article_link']}")
        print(f"Article title: {block['article_title']}")
        print("Article body:")
        print(f"{block['article_body']}\n")
        print(f"Number of articles remaining: {counter}/{totalArticles}")

        # Display categories
        print("\nCategories:")
        print("1. Politics/Government")
        print("2. Business/Economy")
        print("3. Science/Technology")
        print("4. Health/Medicine")
        print("5. Environment/Nature")
        print("6. Culture/Education")
        print("7. Sports/Recreation")
        print("8. Lottery Numbers")
        print("9. Crime/Law")
        print("10. International/Global Affairs \n")
```

Figura 33 - classificação de artigos

Depois de todos os artigos já estarem classificados, o utilizador pode então verificar os resultados, os artigos são divididos aleatoriamente em 30% para testes e 70% para treino, portanto os resultados nunca serão exatamente os mesmos após cada execução do módulo. Colocar aleatoriedade nestes 2 conjuntos permite obter resultados muito mais interessantes do que estar sempre a ver os mesmos artigos a serem utilizados.

A imagem abaixo contém o excerto do código que divide os artigos nos 2 conjuntos.

```python
# Randomly select 70% of the embeddings with data and their categories
num_samples = len(embeddings_with_data)
train_size = int(num_samples * 0.7)
train_indices = set(random.sample(range(num_samples), train_size))
train_embeddings = [embeddings_with_data[i]['embedding'] for i in train_indices]
train_categories = [embeddings_with_data[i]['category'] for i in train_indices]
train_titles = [embeddings_with_data[i]['title'] for i in train_indices]

# Get the remaining 30% of the embeddings with data
test_indices = list(set(range(num_samples)) - train_indices)
test_embeddings = [embeddings_with_data[i]['embedding'] for i in test_indices]
test_categories = [embeddings_with_data[i]['category'] for i in test_indices]
test_titles = [embeddings_with_data[i]['title'] for i in test_indices]

# Calculate cosine similarity between test_embeddings and train_embeddings
cosine_similarities = cosine_similarity(test_embeddings, train_embeddings)

# Calculate Euclidean distances between test_embeddings and train_embeddings
euclidean_dists = euclidean_distances(test_embeddings, train_embeddings)

# Calculate angular distances between test_embeddings and train_embeddings
angular_dists = angular_distances(np.array(test_embeddings), np.array(train_embeddings))
```

Figura 34 - Divisão dos artigos em 30% e 70%

Depois os artigos tem de ser passados para o SBERT para serem reduzidos para 768 fazendo assim com que cada artigo seja transformado num vetor multidimensional. A seguir, só falta chamar cada um dos algoritmos de distância métricas e calcular as novas categorias, abaixo temos o excerto de código que calcula a nova categoria utilizando a similaridade do cosseno (mas não esquecer que cada um dos 4 algoritmos contribui para o cálculo da categoria final). No código abaixo pode-se ver que é atribuído um maior peso se o artigo tiver uma similaridade de 0.8 ou mais, isto porque quanto mais próximo de 1, mais semelhantes os artigos são.

```python
# Create a list to store the cosine similarity results
cosine_results = []

# Find the most similar articles for each article in the test set using cosine similarity
for i, similarities in enumerate(cosine_similarities):
    top_indices = np.argsort(similarities)[::-1][:5]  # Get the indices of the top 5 most similar articles
    top_categories = np.array([train_categories_numerical[idx] for idx in top_indices], dtype=np.float64)  # Convert to NumPy array with dtype=np.float64
    top_similarities = np.array([similarities[idx] for idx in top_indices], dtype=np.float64)  # Convert to NumPy array with dtype=np.float64

    # Create weights based on similarity values
    top_weights = np.array([2 * similarity if similarity >= 0.8 else 0.5 * similarity for similarity in top_similarities], dtype=np.float64)

    # Calculate the weighted average of top categories
    weighted_average = np.average(top_categories, weights=top_weights)

    # Find the new category based on the weighted average
    rounded_average = round(weighted_average)
    new_category = max(category_mapping, key=lambda x: category_mapping[x])  # Initialize with the highest category
    for category, value in category_mapping.items():
        if value == rounded_average:
            new_category = category
            break

    # Create a dictionary to store the result for this test article
    result = {
        "test_article_title": test_titles[i],
        "original_category": test_categories[i],  # Fetch the original category from test_categories
        "weighted_average": weighted_average,
        "new_category": new_category
    }

    # Compare the original category with the new category and update the hit/miss counts
    if result["original_category"] == result["new_category"]:
        cosine_hits += 1
    else:
        cosine_misses += 1

    # Append the dictionary to the list of results
    cosine_results.append(result)
```

Figura 35 - cálculo da categoria usando similaridade de cosseno

No final de todos os cálculos o utilizador irá ver 5 gráficos interativos com as categorias calculadas dos 4 algoritmos de distância métrica, sendo que o 1º gráfico mostra apenas os artigos com a sua categoria original. A imagem abaixo mostra o 1º gráfico interativo exibido após os cálculos das categorias.

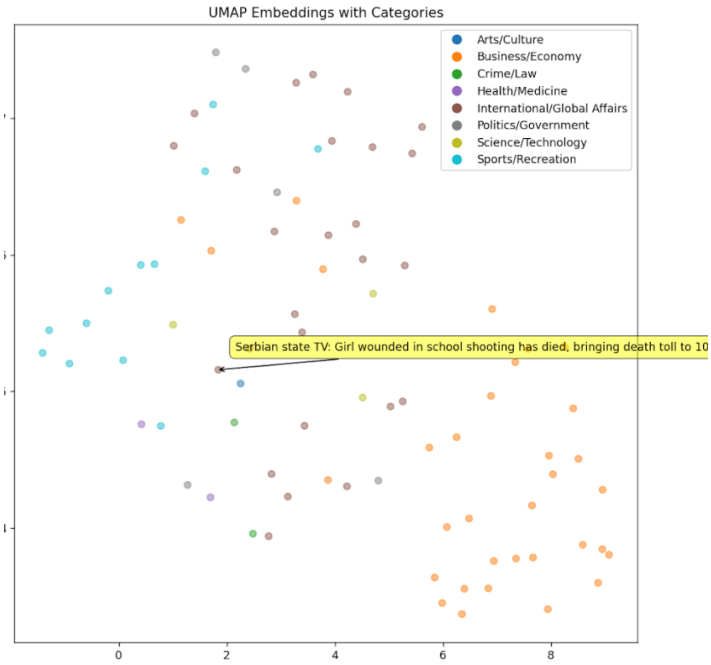


Figura 36 - gráfico com as categorias originais

Por última falta só a funcionalidade mais importante, que é a de comparar o artigo do utilizador com os artigos da base de dados.

Isto é feito após passar os artigos da base de dados e do utilizador novamente pelo SBERT e todos serem reduzidos para 768 dimensões, depois corre-se os 4 algoritmos de distância métrica e comparasse o artigo do utilizador com todos os da base de dados. Cada algoritmo produz uma lista com os 5 artigos mais semelhantes e depois escolhe-se o que tiver a média da soma de cada score de cada algoritmo.

O código abaixo mostra os 4 algoritmos a gerarem cada um a sua lista dos 5 artigos que consideram ser mais semelhantes e depois o cálculo feito para devolver a resposta final ao utilizador.

```
# retrieves the normalized bodies of all articles from the database
trainArticles = list(collection.find({}, {"_id": 0, "article_title": 1, "article_link": 1, "normalized_body": 1}))

# converts the user's article to a 768-dimensional embedding using SBERT
userEmbedding = model.encode([lemmatizedArticle])[0]

# converts the train articles to embeddings using SBERT
trainEmbeddings = model.encode([article["normalized_body"] for article in trainArticles])

# calculates similarity between the user's article and the articles in the database
cosineSimilarity, angularDistances, euclideanDistances, minkowskYDistances = calculateSimilarity(userEmbedding, trainEmbeddings)

# finds the indices of the top 5 most similar articles for each distance measure
cosineTopIndices = np.argsort(cosineSimilarity)[::-1][:5]
angularTopIndices = np.argsort(angularDistances)[:5]
euclideanTopIndices = np.argsort(euclideanDistances)[:5]
minkowskiTopIndices = np.argsort(minkowskYDistances)[:5]

# gets the titles, links, and bodies of the top 5 most similar articles for each distance measure
cosineTopArticles = [{"article": trainArticles[int(idx)], "similarity": float(similarity)} for idx, similarity in zip(cosineTopIndices, cosineSimilarity[cosineTopIndices])]
angularTopArticles = [{"article": trainArticles[int(idx)], "similarity": float(similarity)} for idx, similarity in zip(angularTopIndices, angularDistances[angularTopIndices])]
euclideanTopArticles = [{"article": trainArticles[int(idx)], "similarity": float(similarity)} for idx, similarity in zip(euclideanTopIndices, euclideanDistances[euclideanTopIndices])]
minkowskiTopArticles = [{"article": trainArticles[int(idx)], "similarity": float(similarity)} for idx, similarity in zip(minkowskiTopIndices, minkowskYDistances[minkowskiTopIndices])]

# sorts the articles by highest to lowest similarity
cosineTopArticles = sorted(cosineTopArticles, key=lambda x: x["similarity"], reverse=True)
angularTopArticles = sorted(angularTopArticles, key=lambda x: x["similarity"])
euclideanTopArticles = sorted(euclideanTopArticles, key=lambda x: x["similarity"])
minkowskiTopArticles = sorted(minkowskiTopArticles, key=lambda x: x["similarity"])

# finds the common articles in all four calculations
commonResults = []
cosineCommon = [article["article"]["article_link"] for article in cosineTopArticles]
angularCommon = [article["article"]["article_link"] for article in angularTopArticles]
euclideanCommon = [article["article"]["article_link"] for article in euclideanTopArticles]
minkowskyCommon = [article["article"]["article_link"] for article in minkowskiTopArticles]

commonLinks = list(set(cosineCommon) & set(angularCommon) & set(euclideanCommon) & set(minkowskyCommon))

for link in commonLinks:
    cosineResults = next(item for item in cosineTopArticles if item["article"]["article_link"] == link)["similarity"]
    angularResults = next(item for item in angularTopArticles if item["article"]["article_link"] == link)["similarity"]
    euclideanResults = next(item for item in euclideanTopArticles if item["article"]["article_link"] == link)["similarity"]
    minkowskiResults = next(item for item in minkowskiTopArticles if item["article"]["article_link"] == link)["similarity"]
    averageResults = (cosineResults + (1 - angularResults) + (1 - euclideanResults) + (1 - minkowskiResults)) / 4  # Average of adjusted scores
    commonResults.append({"article_link": link, "average_score": averageResults})

# sorts the articles by highest average score
orderedArticles = sorted(commonResults, key=lambda x: x["average_score"], reverse=True)
```

Figura 37 - comparar artigo do utilizador com a base de dados blockchain

Capítulo 5.4. Abrangência

Neste capítulo vai ser descrito quais são os limites do projeto e que aspetos foram considerados mas não implementados. Vai também analisar possíveis implementações ou melhorias que poderão dar continuidade ao projeto.

Capítulo 5.4.1. Limites do projeto

Começando então pelos limites do projeto, este projeto não se destina ao âmbito comercial, trata-se de um projeto académico para avaliar se os algoritmos de distâncias métricas podem ou não contribuir para o combate às notícias falsas em que os artigos estão armazenados numa blockchain.

A implementação da Blockchain torna impraticável modificar à força o conteúdo da base de dados sem que o programa o detete, mas este projeto é limitado, e portanto não houve tempo de desenvolver mecanismos para guardar a hash da BD, estando a mesma exposta e visível num ficheiro de texto na pasta do projeto, mas ainda assim seria preciso modificar o código para que fosse possível modificar o conteúdo da base de dados à força, o que demonstra o nível de segurança que uma blockchain traz mesmo não tendo a hash devidamente guardada numa localização externa.

Tratando-se de um projeto académico e como foi preciso dedicar a maioria do tempo à implementação da Blockchain e dos algoritmos de distância métrica, também não foi feito a divisão das permissões do utilizador e do administrador, tendo ambos acesso a funcionalidades que deviam estar fora do alcance de ambos (como por exemplo apagar um bloco da blockchain).

A implementação de uma interface gráfica também não foi considerada uma prioridade e não foi implementada.

O critério de seleção dos algoritmos de distância métrica foi que tinham que ser compatível com os resultados que o SBERT produz, que reduz os textos a vetores multidimensionais. O SBERT é capaz de capturar o significado semântico de frases utilizando modelos pré-treinados para tal, produz vetores multidimensionais de dimensão fixa que capturam o contexto da informação e a relação semântica entre as palavras. A partir dai foi só selecionar os 4 algoritmos mais populares e construir módulos com eles para depois observar e por fim combinar os resultados de todos eles.

Apesar de não aparecer é uma limitação, porque automaticamente fiquei limitado a que algoritmos posso utilizar, uma vez que tinham que ser compatíveis com os resultados que o SBERT gerava, ou seja algoritmos como “Named Entity Recognition (NER)”, “Statistical Machine Translation (SMT)” ou “Neural Machine Translation (NMT)” estavam fora de questão porque não funcionam com o SBERT.

Capítulo 5.4.2. Continuidade do projeto

Este projeto tem uma potência enorme para ser continuidade, para já a segurança da Blockchain pode ser substancialmente melhorada ao armazenar a hash num local seguro. Poderá também ser adicionado um mecanismo de "salt" para reforçar ainda mais a segurança.

A blockchain deveria ser distribuída por uma rede de participantes (que podem ser simulados) para a descentralizar, isto envolve claro a implementação de mecanismos de sincronização, o que aumenta exponencialmente a sua robustez. O passo imediatamente a seguir seria implementar um mecanismo de consenso para se adicionar um novo bloco à Blockchain. Ainda dentro do tópico de transparência, a distribuição da Blockchain vai de encontro aos objetivos de transparência da mesma.

Outras possíveis melhorias poderiam ser claro a separação dos perfis do utilizador e do administrador, a implementação de uma interface gráfica e a adição de outros algoritmos que reforçassem a capacidade do projeto encontrar o artigo correto quando o utilizador inserisse o seu.

Por fim creio que seria interessante ligar todo o projeto a uma extensão web que verificasse o conteúdo da página e fosse capaz de devolver uma resposta ao utilizador sobre a veracidade do que está a ler naquele momento.

Capítulo 6. Método e Planeamento

O desenvolvimento deste projeto pode ser divido em essencialmente 3 fases:

-Criação e armazenamento da Blockchain, que como o nome indica consiste na programação da Blockchain e nos respetivos testes para garantir que os blocos estão devidamente ligados uns aos outros e depois armazená-los no mongoDB. Esta fase durou deste o Novembro até Janeiro

Este tempo foi usado para testar várias frameworks e pedaços de código para ver qual melhor se adaptava ao projeto, uma vez que a maioria dos projetos de Blockchain foi pensado para manter cripto moedas precisou de várias modificações. O resto do tempo previsto foi utilizado para testar a inserção da Blockchain na base de dados.

- Programar o extrator de artigos (web scraper). Consistiu na programação dos módulos que recebem um ou mais artigos e que irá extrair apenas o conteúdo essencial da página web para depois esta informação ser inserida na blockchain, mantendo assim apenas a informação necessária. Esta fase durou desde o início de janeiro até ao fim do mês de fevereiro.

- Programar os algoritmos de distância métrica. Consistiu na programação de todos os algoritmos que fazem a comparação dos artigos, dos gráficos interativos e os respetivos resultados. Foia tarefa mais complicada, e durou desde o início do mês de março até ao fim do mês de junho.

- Escrever o relatório final. Como indicado pelo próprio nome, será o tempo necessário para escrever o último relatório e organizar um repositório apresentável aonde todo o código do projeto será disponibilizado para a apresentação final. Esta última fase deverá ir desde o início do mês de junho até à data final da entrega do relatório.

As datas do relatório que antecede este são apresentadas no gráfico Gantt como mostrado na imagem abaixo.

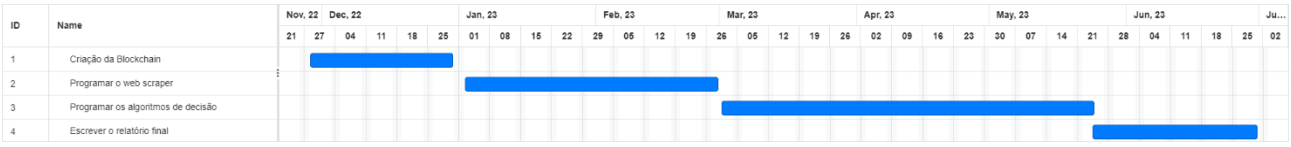


Figura 38 - Cronograma do projeto

Agora que o parte técnica do projeto está concluída pode-se fazer uma análise sobre a estimativa e datas apresentadas na imagem acima que tem as datas estimadas no relatório que antecede este.

Como tal pode-se ver que a única etapa que se prolongou mais do que estava inicialmente previsto foi a implementação dos algoritmos de distância métrica. Inicialmente não estava previsto nenhum sistema de classificação dos artigos e os gráficos interativos, tendo isso sendo um requisito do orientador técnico, sendo essa a causa de o projeto não estar concluído a tempo da 1ª entrega.

Em retrospetiva o sistema de classificação pedido é na verdade bastante útil para analisar resultados sobre a eficácia de cada um dos algoritmos implementados. Tanto mais que o modulo que compara o artigo do utilizador com todos os artigos da base de dados para determinar o mais semelhante é apenas um reaproveitamento do código que divide os artigos da base de dados em 30%/70% para analise.

Tirando esse pedido que não estava comtemplado no plano original do projeto, todos os objetivos propostos foram alcançados.

Capítulo 7. Resultados

Como já foi indicado anteriormente, todos os requisitos inicialmente propostos foram alcançados e até as propostas adicionais do sistema de classificações foram devidamente implementadas.

Os requisitos iniciais era a implementação de uma estrutura de blockchain, um mecanismo para extrair o conteúdo de páginas web e uma maneira de comparar o artigo inserido do utilizador com o conteúdo armazenado. O design inicial do projeto consistia apenas nestas funcionalidades e todas elas foram implementadas com sucesso. As únicas funcionalidades que acabaram por ser implementadas de forma diferente do que foi originalmente pensado foram:

- Extrair conteúdo de páginas web, no início do projeto foi inicialmente pensado que seria muito mais fácil do que realmente é extrair o conteúdo do artigo de qualquer página web, mas como já foi dito anteriormente isso é uma funcionalidade praticamente impossível de se implementar.

- Extrair automaticamente do URL do utilizador o corpo do artigo, também não foi possível implementar esta funcionalidade exatamente pelas razoes descritas no ponto anterior.

Durante o desenvolvimento do projeto foi-me aconselhado a desenvolver um sistema de classificação de artigos e apesar de não estar inicialmente prevista no projeto foi implementada com sucesso. É esta mesma funcionalidade que permite tirar conclusões sobre a fiabilidade dos algoritmos implementados.

Capítulo 7.1. Resultados dos algoritmos implementados

Este capítulo destina-se única e exclusivamente a demonstrar os resultados obtidos na classificação de artigos utilizando os 4 algoritmos de distância métrica implementados.

Estes algoritmos são utilizados para comparar o corpo do artigo que o utilizador inseriu com todos os artigos da base de dados blockchain.

O processo utilizado é igual ao usado para comparar as categorias, exceto que invés de devolver uma categoria, é devolvido o que o programa considera ser o artigo mais semelhante. O processo é o seguinte:

O código começa por receber um artigo inserido pelo utilizador.

Esse artigo é convertido num vetor de 768 dimensões usando o modelo SBERT (Sentence-BERT), o qual é uma técnica de processamento de linguagem natural para representar o texto em formato numérico.

Em seguida, o código calcula quatro métricas de similaridade entre o vetor do artigo inserido pelo utilizador e os vetores de outros artigos existentes na base de dados. As métricas utilizadas são a similaridade do cosseno, a distância angular, a distância euclidiana e a distância de Minkowski.

Para cada métrica de similaridade, o código seleciona os 5 artigos mais similares em relação ao artigo inserido pelo utilizador. Esses 5 artigos são armazenados em listas separadas para cada métrica.

Em seguida, o código procura os artigos que aparecem em todas as listas dos 5 mais similares, ou seja, aqueles que são comuns a todas as métricas de distância. Estes artigos são considerados os mais relevantes e consistentes nas diferentes medidas de similaridade.

Para os artigos comuns, o código calcula a média da similaridade com base nas pontuações obtidas em cada métrica de distância.

Os artigos comuns são, então, ordenados com base na média de similaridade calculada. Dessa forma, os artigos com as maiores médias de similaridade aparecem primeiro na lista.

Finalmente, o código apresenta ao utilizador o artigo com a maior média de similaridade e mostra a pontuação média alcançada. Este é o artigo considerado o mais semelhante ao artigo inserido pelo utilizador, de acordo com as diferentes métricas de similaridade utilizadas no processo.

Em baixo temos alguns exemplos dos resultados obtidos, cada resultado consiste num link de um artigo que se encontra na base de dados e um link obtido na internet ao procurar pelo tema do artigo, cada resultado terá uma imagem do resultado do projeto e descrição se é o resultado pretendido. Assume-se por resultado pretendido se o programa devolver o link original que se encontra na base de dados como sendo o artigo mais semelhante.

(Para permitir reproduzir estes resultados, esta base de dados está na pasta do projeto no repositório Git e pode ser colocada no projeto utilizando a opção '12' do menu e depois invocando a opção '2' para inserir corpos de artigos.)

Caso 1

Link Original na BD:

https://apnews.com/article/alaska-ranked-choice-voting-5ae6c163af2f8a70a8f90928267c4086

Conteúdo comparado com:

https://www.nytimes.com/2022/11/08/us/politics/ranked-choice-voting.html

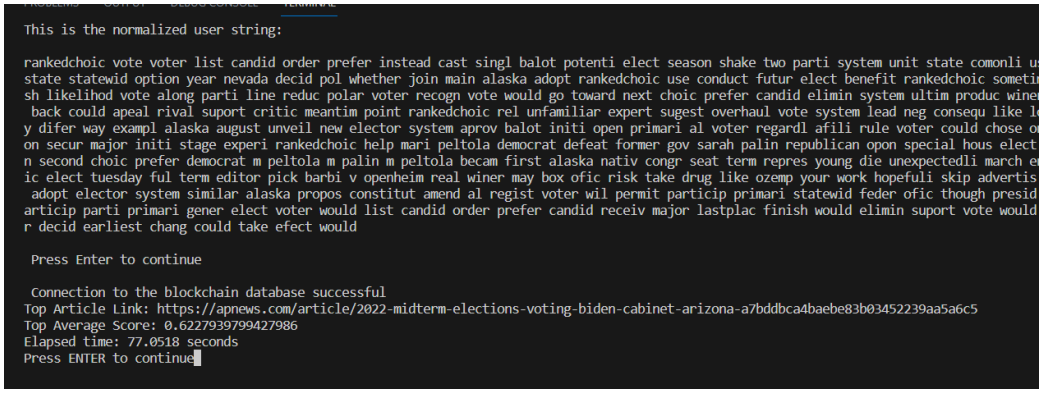


Figura 39 - Resultado não pretendido caso 1

Como se pode ver o link devolvido pelo programa não é o correto. Como tal foi realizado outro teste com outro artigo sobre o mesmo tema. Mas desta utilizando este link https://edition.cnn.com/2022/08/31/politics/alaska-how-ranked-choice-voting-works/index.html, o resultado obtido foi:

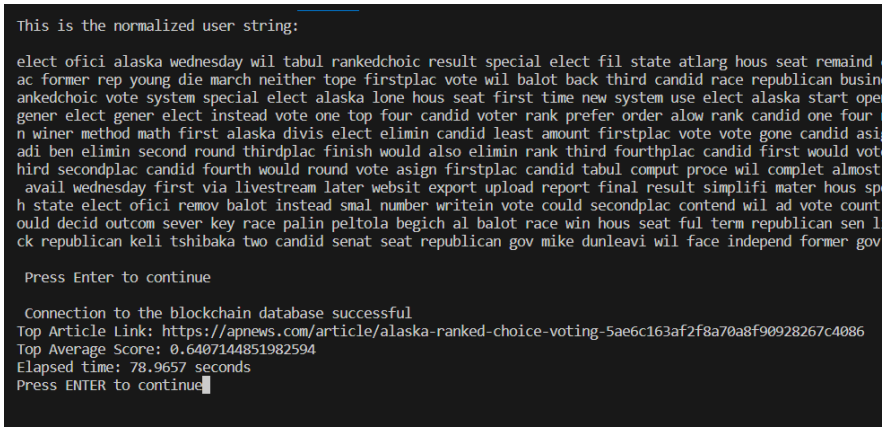


Figura 40 - Resultado pretendido caso 1

A imagem 40 mostra de facto o artigo que se encontra na base de dados blockchain a ser devolvido como sendo o artigo mais semelhante encontrado, sendo esse o resultado esperado.

Em baixo encontra-se uma tabela com outros testes feitos.

Link original	Link comparado(s)	Resultado	Score
midterm-elections-lawsuits-arizona-phoenix	rep-ryan-zinke-endorses-tim-sheehy-for-montana-senate	midterm-elections-lawsuits-arizona-phoenix	0.6404
	trump-interior-sec-ryan-zinke-wins-congressional-state-in-montana		0.6326
elon-musk-technology-misinformation-nancy-pelosi	elon-musk-twitter-baseless-conspiracy-theory-paul-pelosi-attack	earnings-bb469de85c9e21ca0eae8d54fe5a00d8	0.6154
	elon-musk-tweets-conspiracy-theory-about-pelosi-attack	midterm-elections-elon-musk-technology-misinformation-nancy-pelosi-	0.7356
health-business-china-covid	china-imposes-covid-lockdown-on-area-around-iphone-factory	health-business-china-covid	0.6626
	China closes zone around iPhone factory after virus cases	floods-pakistan-weather-climate-and-environment	0.6250
nhl-sports-arizona-florida-hockey	arizona-coyotes-defeat-florida-panthers-3-1-for-first-win-at-mullett-arena	nhl-sports-hockey-ontario-canada	0.6972
	matthew-tkachuk-coyotes-panthers	college-football-sports-south-carolina-gamecocks-shane-beame	0.6658

Como se pode ver pela tabela acima, o programa é capaz de encontrar o link pretendido, mas também existem várias vezes em que não consegue mesmo encontrar o link correto.

Para estes testes foram excluídos links cujos artigos pareciam ser do mesmo tópico que um link que se encontrava naquele momento na base de dados.

O código pode retornar resultados esperados em alguns casos e em outros não, devido a várias razões e limitações dos algoritmos de métricas de distância utilizados. Eis alguns fatores a considerar e conclusões que se podem tirar:

Qualidade dos Artigos: A eficácia das métricas de similaridade depende da qualidade dos artigos geradas pelo modelo SBERT. Se o SBERT não conseguir capturar corretamente a essência dos artigos, as métricas de similaridade serão menos confiáveis.

Escolha das Métricas de Distância: Diferentes métricas de distância têm as suas vantagens e limitações. Por exemplo, a similaridade do cosseno é robusta à magnitude dos vetores, mas pode não lidar bem com algumas nuances dos dados. Por outro lado, a distância euclidiana é sensível à magnitude, o que pode afetar os resultados se os vetores tiverem escalas diferentes.

Limitações do SBERT: O modelo SBERT, embora poderoso, pode ter dificuldades com certos tipos de artigos ou línguas. Pode ter dificuldade em capturar o contexto e a semântica em textos altamente técnicos ou específicos de um domínio.

Base de Dados Pequena: A qualidade dos resultados de similaridade também é influenciada pelo tamanho e diversidade da base de dados. Se a base de dados contiver poucos artigos ou abranger um número limitado de tópicos, poderá não fornecer recomendações precisas.

Subjetividade da Similaridade: A noção de "similaridade" pode ser subjetiva. Diferentes métricas de distância podem enfatizar diferentes aspetos da similaridade, e o que é considerado similar por uma métrica pode não corresponder ao julgamento humano.

Impacto da Normalização: Embora a normalização possa melhorar a qualidade dos cálculos de similaridade, ela pode não ser apropriada para todos os conjuntos de dados e pode introduzir viés.

Utilização com Outros Sistemas: Apesar dessas limitações, os algoritmos de métricas de distância podem ser valiosos quando usados em conjunto com outros sistemas ou métodos. Ao combinar várias métricas de similaridade e, potencialmente, incorporar o feedback do utilizador, é possível criar um mecanismo de pontuação de similaridade mais robusto e preciso.

Abordagens de Conjunto: Abordagens de conjunto que combinam os resultados de várias métricas de distância ou outros algoritmos podem fornecer melhores recomendações. Ao

obter um consenso de vários métodos, o sistema pode melhorar o seu desempenho geral.

Filtragem Colaborativa: A filtragem colaborativa, uma técnica comum em sistemas de recomendação, também pode ser utilizada para melhorar os cálculos de similaridade, aproveitando as interações e preferências do utilizador.

Melhorias Específicas de Domínio: Para domínios ou tópicos específicos, personalizar o cálculo de similaridade, considerando características ou particularidades específicas do domínio, pode melhorar a precisão dos resultados.

Em conclusão, apesar das limitações dos algoritmos de métricas de distância, eles ainda podem ser úteis em combinação com outros sistemas e técnicas para encontrar os artigos mais corretamente similares. Ao usar abordagens de conjunto, incorporar o feedback do utilizador e adaptar os cálculos a domínios específicos, é possível mitigar essas limitações e alcançar recomendações mais precisas. Além disso, experimentar diferentes métricas de distância e parâmetros pode ajudar a identificar a melhor abordagem para uma determinada aplicação.

Bibliography

[1] “Explained: What is False Information (Fake News)?,” webwise, [Online]. Available: https://www.webwise.ie/teachers/what-is-fake-news/. [Acedido em 22 11 2022].

[2] “Fighting Fake News with Blockchain,” Eidosmedia, 09 05 2022. [Online]. Available: https://www.eidosmedia.com/blog/technology/Fighting-fake-news-with-blockchain. [Acedido em 17 11 2022].

[3] A. Kanski, “study: 86% of people don't fact check news spotted on social media,” PR Week, [Online]. Available: https://www.prweek.com/article/1431578/study-86-people-dont-fact-check-news-spotted-social-media. [Acedido em 15 11 2022].

[4] C. A. D. C. D. R. R. S. A. F. D. P. S. R. Q. L. HUMBERTO JORGE DE MOURA COSTA, “A Fog and Blockchain Software Architecture for a Global Scale Vaccination Strategy,” IEEE, 21 march 2022. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9761908. [Acedido em 24 01 2023].

[5] S. Koren, “Introducing the News Provenance Project,” NYT Open, 23 06 2019. [Online]. Available: https://open.nytimes.com/introducing-the-news-provenance-project-723dbaf07c44. [Acedido em 18 11 2022].

[6] W. i. Identity, “Kathryn Harrison: Deepfakes and the Deep Trust Alliance,” Women in Identity, [Online]. Available: https://www.womeninidentity.org/articles/kathryn-harrison-deepfakes-and-the-deep-trust-alliance. [Acedido em 20 11 2022].

[7] ibm, “Safe.press,” IBM, [Online]. Available: https://www.ibm.com/downloads/cas/6D0XAJQN. [Acedido em 17 11 2022].

[8] A. D. Waghmare, “FAKE NEWS DETECTION OF SOCIAL MEDIA NEWS IN BLOCKCHAIN FRAMEWORK,” IJCSE, 4 07 2021. [Online]. Available: http://www.ijcse.com/docs/INDJCSE21-12-04-151.pdf. [Acedido em 22 11 2022].

[9] “ANSA leveraging blockchain technology to help readers check source of news,” ansa, 06 04 2020. [Online]. Available: https://www.ansa.it/english/news/science_tecnology/2020/04/06/ansa-using-blockchain-to-help-readers_af820b4f-0947-439b-843e-52e114f53318.html.

[10] FACT, “Fact Protocol (FACT) - White Paper Draft,” FACT, 2022. [Online]. Available: https://fact.technology/files/2022/05/Fact-Protocol-White-Paper-Q2-2022-signed.pdf.

[11] O. Harrison, “Machine Learning Basics with the K-Nearest Neighbors Algorithm,” towardsdatascience, 2018. [Online]. Available: https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761#:~:text=KNN%20works%20by%20finding%20the,in%20the%20case%20of%20regression)..

[12] “Logistic Regression For Machine Learning and Classification,” kambria, 2019. [Online]. Available: https://blog.kambria.io/logistic-regression-for-machine-learning/.

[13] T. Joachims, “Text Categorization with Support Vector Machines: Learning with many relevant features,” Universit□at Dortmund, [Online]. Available: https://www.cs.cornell.edu/people/tj/publications/joachims_98a.pdf.

[14] “Naïve Bayes Classifier Algorithm,” javapoint, [Online]. Available: https://www.javatpoint.com/machine-learning-naive-bayes-classifier.

[15] M. T. E. Nabil, “Proof of Credibility: A Blockchain Approach for Detecting and Blocking Fake News in Social Networks,” IJACSA, 2019. [Online]. Available: https://thesai.org/Downloads/Volume10No12/Paper_43-Proof_of_Credibility_A_Blockchain_Approach.pdf. [Acedido em 21 11 2022].

[16] “what is blockchain,” money.com, [Online]. Available: https://money.com/what-is-blockchain/. [Acedido em 23 11 2022].

[17] pcmag, “what is the blockchain and whats it used for,” [Online]. Available: pcmag.com/how-to/what-is-the-blockchain-and-whats-it-used-for. [Acedido em 19 11 2022].

[18] b. academy, “What is a 51% Attack?,” [Online]. Available: https://academy.bit2me.com/en/which-is-a-51-attack/. [Acedido em 22 11 2022].

[19] E. Costa, “The Benefits and Vulnerabilities of Blockchain Security,” cengn.ca, 19 09 2021. [Online]. Available: https://www.cengn.ca/information-centre/innovation/the-benefits-and-vulnerabilities-of-blockchain-security/. [Acedido em 11 11 2022].

[20] “Can blockchain block fake news and deep fakes?,” IBM, 30 11 2020. [Online]. Available: https://www.ibm.com/blogs/industries/blockchain-protection-fake-news-deep-fakes-safe-press/. [Acedido em 15 11 2022].

[21] “Fake News, Disinformation, and Deepfakes: Leveraging Distributed Ledger Technologies and Blockchain to

Combat Digital Deception and Counterfeit Reality," CITIC, 20 10 2019. [Online]. Available: https://arxiv.org/pdf/1904.05386.pdf. [Acedido em 10 11 2022].

[22] A. Leopold, "How Blockchain Can Help Combat Disinformation," Hardvard Bussiness Review, 19 07 2021. [Online]. Available: https://hbr.org/2021/07/how-blockchain-can-help-combat-disinformation. [Acedido em 18 11 2022].

[23] G. MILLER, "As U.S. election nears, researchers are following the trail of fake news," Science.org, 26 09 2020. [Online]. Available: https://www.science.org/content/article/us-election-nears-researchers-are-following-trail-fake-news. [Acedido em 20 11 2022].

[24] n. subedi, "FastText: Under the Hood," 7 7 2018. [Online]. Available: https://towardsdatascience.com/fasttext-under-the-hood-11efc57b2b3.