Uma Ferramenta para Encontrar Similaridade entre Notícias de Websites Brasileiros[1]

João Antonio Leite Dos Santos Neto, Yuri de Almeida Malheiros Barbosa

Departamento de Ciências Exatas - Universidade Federal da Paraíba (UFPB)

{joao.antonio, yuri}@dcx.ufpb.br

Abstract. *Nowadays there is a huge diversity of sources of information accessible to the population, as a consequence, problems such as fake news and information bias, compromise the quality of the news. The reader then looks for consolidated sites that sell credibility. This work aims to develop a tool using the technique Latent Semantic Index(LSI), that suggests according to a notice, others of the same subject by collecting data from the three most accessed news portals in Brazil according to the Alexa ranking (2018).*

Resumo. *Existe hoje uma enorme diversidade de fontes de informação acessíveis à população, como consequências, problemas como fake news e enviesamento da informação, compromete a qualidade das notícias. O leitor procura então websites consolidados que venda credibilidade. Este trabalho tem como propósito desenvolver uma ferramenta, utilizando a técnica Latent Semantic Index(LSI), que sugere de acordo com uma notícia, outras do mesmo assunto, através da coleta de dados dos três portais de notícias mais acessados do Brasil de acordo com o ranking da Alexa (2018) .*

1. Introdução

Devido aos avanços da Internet, hoje se tem uma enorme diversidade de fontes publicando suas notícias em websites acessíveis à população. No entanto, cada website possui seu viés o que tende a levar à uma visão tendenciosa e particular de um contexto. Em sua pesquisa [COUTO 2013], analisando a forma como os portais UOL e Terra descreveram as mortes de Hugo Chávez e Margaret Thatcher, confirmaram sua hipótese de que as notícias relacionadas à vida política de Hugo Chávez são retratadas com viés negativo, enquanto o governo e postura de Thatcher como positivos.

Além disso, nos últimos anos o crescimento de notícias fraudulentas tem manipulado os usuários da rede mundial de computadores. A viralização de informações denegrindo a imagem da vereadora Marielle Franco, assassinada no Rio de Janeiro em março deste ano, bem como a disseminação de 2,1 milhões de publicações através da plataforma Twitter, facilitaram a propagação de fatos falsos manipulando o julgamento de parte da população sobre o caso [ALMEIDA 2018]. Desta forma, a veracidade é comprometida a partir do momento que relatos incorretos são repassados. Em seu estudo [VOSOUGHI 2018] afirma que uma notícia falsa possui chances de ser divulgada mais rapidamente do que uma verdadeira e que isto acontece principalmente nas relacionadas à política.

[1] Trabalho de Conclusão de Curso (TCC) na modalidade Artigo apresentado como parte dos pré-requisitos para a obtenção do título de Bacharel em Sistemas de Informação pelo curso de Bacharelado em Sistemas de Informação do Centro de Ciências Aplicadas e Educação (CCAE), Campus IV da Universidade Federal da Paraíba, sob a orientação do professor sob orientação do professor Yuri de Almeida Malheiros Barbosa.

A necessidade de fontes confiáveis no ambiente virtual atual é de grande importância visto que as pessoas estão cada vez mais receosas na qualidade das notícias procurando portais consolidados que fornecem credibilidade. No entanto, do ponto de vista do leitor, uma imensa quantidade de informações são expostas diariamente inviabilizando a pesquisa em diversas fontes sobre um mesmo assunto. Além do tempo e esforço necessário, escolhas de títulos semanticamente diferentes dificultam a procura. Como consequência acaba obtendo-se como referência a imparcialidade de um determinado portal sem diferentes perspectivas e opiniões, impossibilitando o julgamento de diferentes pontos de vista.

Este artigo apresenta uma ferramenta que tem como objetivo sugerir de acordo com uma notícia, outras do mesmo assunto de outros portais, auxiliando o leitor a obter múltiplas visões sobre uma mesma informação, comparando a forma como cada um conta uma mesma história. Para o desenvolvimento da ferramenta, foi utilizado o LSI (*Latent Semantic Index*), uma técnica que permite sugerir dado um texto, outros textos similares através da comparação utilizando representação vetorial [WIEMER-HASTINGS 2004] .

2. Referencial Teórico

2.1. Processamento de Linguagem Natural

O processamento de linguagem natural (PLN) dedica-se aos aspectos da comunicação humana como sons, palavras, sentenças e discursos, com o objetivo de encontrar uma solução estruturada para interpretar o dialeto de maneira computacional. Em resumo, o PLN propõe-se a compreender a linguagem humana através do processamento em níveis diferentes de entendimento: fonético e fonológico, morfológico, sintático, semântico e pragmático [GONZALES 2003]. É abordado neste artigo diretamente o nível relacionado à semântica, pois faz-se necessário a interpretação do texto pertencente a uma notícia através da análise de discursos com o propósito de alcançar a interpretação de sentenças inseridas no contexto, considerando as antecessoras como fortes influentes nas frases sucessoras [BALDAN 2012].

Uma abordagem muito utilizada para a representação de documentos é o modelo *bag-of-words* que é uma representação estruturada para a contagem do número de ocorrências de termos. O documento é transformado em um conjunto de palavras, atribuindo para cada uma seu respectivo peso (número de ocorrências). Por exemplo, dados os documentos a seguir:

- *'Interface homem-máquina para aplicações de computador de laboratório'.*
- *'Uma pesquisa da opinião do usuário do tempo de resposta do sistema de computador'.*
- *'O EPS sistema de gerenciamento de interface do usuário'.*
- *'Teste de engenharia de sistema e sistema humano de EPS'.*

Determina-se um identificador para cada palavra do *corpus* dessa forma :

> *{'aplicações': 0, 'computador': 1, 'homem-máquina': 2, 'interface': 3, 'laboratório': 4, 'opinião': 5, 'pesquisa': 6, 'resposta': 7, 'sistema': 8, 'tempo': 9, 'usuário': 10, 'eps': 11, 'gerenciamento': 12, 'engenharia': 13, 'humano': 14, 'teste': 15 }.*

Assim, os documentos anteriores podem ser representados através de uma lista de pares, nos quais o primeiro elemento é o identificador da palavra e o segundo é a frequência da palavra no documento. O propósito é reduzir a alta dimensionalidade, pois termos iguais em documentos diferentes são representados pelo mesmo identificador. Os documentos passam a ter a seguinte representação no modelo *bag-of-words*:

- *(0, 1), (1, 1), (2, 1), (3, 1), (4, 1),*
- *(1, 1), (5, 1), (6, 1), (7, 1), (8, 1), (9, 1), (10, 1)*
- *(3, 1), (8, 1), (10, 1), (11, 1), (12, 1)*
- *(8, 2), (11, 1), (13, 1), (14, 1), (15, 1)*

A utilização do modelo *bag-of-words* acarreta em consequências como perda da ordenação das palavras e desconsideração da gramática para atribuir os pesos [GEORGE 2014]. Por exemplo, o termo "computador" com id "1" que estava no final da primeira frase se encontra na segunda posição do array.

No PLN, as *stopwords* são palavras comuns que não agregam significado a um documento [SOLKA 2008]. É possível identificar algumas delas nos textos dos documentos do exemplo anterior: "de", "da", "do", "e" e "para". A remoção de tais palavras permite a redução do número de palavras nos documentos sem a perda semântica dos textos. Um dos principais problemas da utilização da abordagem *bag-of-words* está na valorização de palavras que aparecem na maior parte dos documentos. Esses termos genéricos impactam negativamente na análise de similaridade. Isto ocorre devido à possível classificação de documentos semanticamente distintos serem taxados como semelhantes.

O modelo TF-IDF (*Term frequency - inverse document frequency*) é uma técnica utilizada para a redução de possíveis ruídos provocados pelo *bag-of-words*. A ideia é converter vetores de números inteiros em real sem alterações no número de dimensões, aplicando uma ponderação que valoriza palavras específicas do documento e desvaloriza as comuns ao *corpus*.

É aplicado o princípio no qual palavras que aparecem muitas vezes no texto e no *corpus* são penalizadas. Já as palavras que aparecem com muita frequência no texto e poucas vezes no *corpus* recebem um valor maior. Para o cálculo do TF-IDF, são aplicadas as seguintes equações:

(1) $$tf(t_j, d_i) = freq(t_j, d_i)$$

(2) $$idf(t_j) = \log \frac{N}{d(t_i)}$$

(3) $$tfidf(t_j, d_i) = freq(t_i, d_j) \ . \ idf(t_j)$$

Na equação (1), *Term frequency - $tf(t_j,d_i)$,* a frequência do termo t_j é dada pelo número de ocorrências no documento d_i. Esta equação é obtida através da aplicação da técnica *bag-of-words*.

Os valores são ajustados pela equação (2) *Inverse document frequency - $idf(t_j)$,* onde N é o número de documentos dentro do *corpus* e $d(t_j)$ é o número de documentos

onde o termo t_j aparece no mínimo uma vez [MATSUBARA 2003]. Dessa forma, é decrementado o peso dos termos comumente usados e aumenta o peso do termos que são pouco usados no *corpus*.

A técnica TF-IDF, ilustrada na equação (3), é o resultado da multiplicação das equações (1) e (2). Dessa forma, é obtida, para cada notícia, a frequência de um termo ajustado para o quão raramente ele é utilizado. A partir do modelo resultante, é possível quantificar de uma forma consistente a importância de termos em um documento.

2.2. LSI

O modelo LSI (*Latent Semantic Indexing ou Indexação Semântica Latente*) é uma técnica elaborada para a tarefa de Recuperação de Informação, selecionando de um grupo de documentos, os mais relevantes a consulta desejada. Nela, é utilizada abordagens desenvolvidas anteriormente que propõem a ponderação de palavras e representação baseadas em uma matriz de termos-documentos [WIEMER-HASTINGS 2014]. Através do LSI é possível relacionar as palavras pertencentes à uma coleção de documentos utilizando a representação através da redução de dimensões do espaço vetorial sem perdas de dados importantes. O propósito é explorar um conjunto implícito de dependências entre termos de acordo com contexto inserido [SILVA 2011]. Cada documento é transformando em um pequeno conjunto de tópicos [PAPADIMITRIOU 2000]. Dessa forma, é possível elaborar consultas para obter o maior grau de relevância entre dados com um melhor desempenho.

Aspectos linguísticos como sinonímia em que é necessário identificar que palavras como “casa”, “residência” e “apartamento” representam a mesma ideia de “lugar onde se mora”. Bem como a polissemia no qual é necessário distinguir palavras com múltiplos significados como, por exemplo, “pena”, tornam a recuperação da informação uma tarefa não trivial. Para lidar com tais problemas, é necessário representar documentos não apenas por termos e ponderações (modelos baseados em vetores como *bag-of-word*s e TF-IDF), mas também através de conceitos latentes ou ocultos obtidos por meio dos termos. Para elaborar um mapeamento de termos-conceitos, a técnica LSI depende criticamente de um conjunto de documentos, bem como a correlação entre os termos [PAPADIMITRIOU 2000].

O LSI utiliza da abordagem baseada em vetores usando a técnica de álgebra matricial SVD (*Singular Value Decomposition*) para a reestruturação dos documentos reorientando e classificando as dimensões em um espaço vetorial [WIEMER-HASTINGS 2004]. As dimensões obtidas através do resultado do SVD são ordenadas da mais para menos significante. Isto permite a eliminação de dimensões menos relevantes promovendo a redução do espaço semântico de busca, eliminando o ruído na representação do documento mantendo o resultado semântico [LUKINS 2008]. Esta nova matriz gerada a partir do SVD é uma representação reduzida e compacta da matriz TF-IDF e pode ser utilizada para o cálculo de similaridade entre documentos [SILVA 2011]. O método *Similaridade de Cosseno* calcula o produto de dois vetores medindo o cosseno do ângulo entre eles em um espaço vetorial [ARAÚJO 2012]. É possível medir a similaridade entre dois documentos, ou entre um documento específico com todo o corpus, permitindo a identificação dos mais semelhantes, através da pontuação obtida no cálculo do cosseno do ângulo.

3. Metodologia

Inicialmente foi desenvolvido um rastreador web responsável pela navegação e extração de dados de portais de notícias. Logo em seguida, os textos das notícias coletadas passaram por um pré-processamento, removendo palavras desnecessárias descritas na Seção 3.2. Posteriormente foi executado o algoritmo LSI com o intuito de calcular as semelhanças entre documentos. Em seguida, foi desenvolvida uma ferramenta que agrupa e sugere notícias de grandes portais sobre temas semelhantes para auxiliar o leitor a ter diferentes visões sobre um mesmo assunto. Por fim, foi aplicado um questionário com o propósito de analisar o grau de concordância dos usuários em relação a ponderação retornada pelo modelo criado na Seção 3.3.

3.1. Coleta de Dados

Utilizando o framework *Scrapy*[2], foi desenvolvido um web crawler para navegação e extração dos conteúdos dos três portais de notícias mais acessados no Brasil de acordo com o ranking da Alexa (2018), respectivamente o *Uol*, *G1* e *BlastingNews*. As notícias foram mineradas separadamente, através de três *spiders* criados[3], um para cada website. Foram coletadas no total 188.820 notícias entre 01 de Janeiro à 16 de Outubro de 2018. Dessas notícias, 128.411 são do *Uol*, 39.629 do *BlastingNews* e 20.780 do *G1*. O *G1* aparece com a menor quantidade de notícias, pois só é possível extrair às notícias dos últimos 5 meses. A disparidade do Uol em relação aos demais ocorre devido à quantidade de domínios vinculados ao portal, sendo possível à extração de notícias de diversas fontes como *BBC, DW* e *The New York Times*. O número total de notícias pode aumentar ao longo do tempo, pois novas notícias são inseridas diariamente através da execução de *scripts* automatizados.

O Scrapy possui um agendador que permite à persistência das requisições, possibilitando a identificação de requisições duplicadas. É impedido que notícias iguais sejam cadastradas pois é possível verificar se solicitações à uma determinada url já foi exigida ou não. Além disso, ao executar um script para algum dos websites e por alguma razão seja necessário interrompê-lo, é possível continuar a extração em outro momento.

Os *scripts* foram executados separadamente, realizando requisições concorrentemente para cada website considerado neste trabalho. Os *scripts* podem ser executados quantas vezes forem necessários. Na ferramenta desenvolvida, os *spiders* foram alocados em uma tarefa no sistema operacional em uma máquina virtual para serem executados a cada duas horas com o propósito de manter o banco de dados sempre atualizado. Em seguida, as notícias são salvas no banco de dados do servidor. Para cada notícia são armazenados o título, subtítulo, corpo e data.

3.2. Pré-processamento de dados

As notícias coletadas foram pré-processadas padronizando as palavras em minúsculas, removendo acentos e eliminando caracteres especiais (&[]¸"‹›…¶^/*.). Também foram removidas as *stopwords*. Os dados pré-processados foram salvos em um banco de dados no momento da extração das notícias através dos *spiders*. É evitado então, esforços desnecessários no momento da requisição dos usuários à ferramenta. Além disso, não é necessário a repetição do mesmo pré-processamento a cada nova requisição. Em seguida, é aplicado o modelo *bag-of-words* e TF-IDF.

---

[2] https://scrapy.org/

[3] https://github.com/joaonetoo/scrapy-noticias

3.3. Análise de Similaridade

O principal propósito do trabalho é encontrar as notícias mais semelhantes dada uma outra notícia como entrada. Embora seja possível as informações abordarem diretamente assuntos iguais que estejam afastadas cronologicamente, o escopo é delimitado as diferentes visões sobre uma notícia em um determinado momento. Além disso, a redução do escopo ajuda a melhorar o desempenho da ferramenta, pois o tamanho do *corpus* é menor, consequentemente menos tempo de processamento é exigido.

A data é utilizada para a filtragem, inserindo no *corpus* apenas as que possuem proximidade de sete dias com a notícia de entrada. Por exemplo, se uma notícia possui a data 08 de Maio de 2018 serão retornadas para o usuário, notícias entre os dias 01 de Maio de 2018 à 15 de Maio de 2018. Para datas muito recentes o intervalo é de 7 dias, pois não existem datas superiores. Existe alguns temas como a greve dos caminhoneiros (2018) que paralisou o Brasil durante dez dias, consequentemente repercutindo notícias relacionadas durante muito tempo. No entanto, foge do escopo do trabalho analisar uma curva temporal das notícias. Para a análise de similaridade é passado como parâmetro apenas o corpo, pois muitos portais tende a elaborar títulos e subtítulos sensacionalistas sendo imprecisos quando correlacionados com o texto. Dessa forma, é necessário ignorá-los nas transformações TF-IDF e LSI, porém serão úteis para iniciais investigações de similaridade.

Na figura 1, é possível visualizar todas as etapas necessárias para elaborar o modelo LSI. Inicialmente temos um conjunto de notícias extraídas dos websites, em seguida o texto é pré-processado através da aplicação de técnicas mostradas na Seção 3.2. Na etapa 2 é aplicado a abordagem *bag-of-words* através da contagem de termos obtidos no pré-processamento, contendo para cada termo o identificador e sua frequência de aparições no documento. Logo em seguida é calculado um modelo de espaço vetorial utilizando a técnica TF-IDF passando como parâmetro o vetor *bag-of-word*. Dessa forma, é retornado uma lista multidimensional de tamanho proporcional ao *corpus* com as medidas probabilísticas de todos os termos contidos em cada uma das notícias.

Figura 1. Etapas de desenvolvimento do modelo LSI.

Através da biblioteca *Gensim*[4] é possível utilizar o algoritmo LSI, mostrado na etapa 4 da figura 1, passando como parâmetro o *corpus* TF-IDF e o vocabulário de palavras, obtendo como retorno um modelo LSI, reduzindo as dimensões da matriz sem perdas de dados importantes. Isto permite descobrir um conjunto latente de dependências entre termos contidos no *corpus*.

Uma vez criado o modelo, é gerado um espaço com 100 dimensões. Como a quantidade de notícias do *corpus* é pequena devido à seleção através da proximidade de datas, não foi necessário aumentar a quantidade de dimensões para obter resultados

[4] https://radimrehurek.com/gensim/

satisfatórios de similaridade. Em [DEERWESTER 1990] é discutido que o número de dimensões ideal, propondo alta precisão para uma representação de 1000 à 2000 documentos é de aproximadamente 100 dimensões. Logo, esta quantidade de dimensões pode ser utilizada na pesquisa, pois são retornadas um número aproximado de notícias no intervalo de até 15 dias.

Com o intuito de encontrar as notícias semelhantes, dada uma notícia de entrada, é calculada a *Similaridade de Cosseno* entre a entrada e todas as outras no espaço com dimensões reduzidas pelo LSI. Para isto, é passado como entrada a posição da notícia no corpus, ou, o corpo da notícia que pretende encontrar às similares. Neste segundo caso, o corpo da notícia é pré-processado, utilizando o dicionário criado anteriormente, passando o vetor *bag-of-words* para a matriz TF-IDF e LSI treinados. Após aplicar a *Similaridade de Cosseno*, o resultado é um vetor com tamanho proporcional ao transformado contendo a pontuação para cada posição do *corpus* a distância entre o dado de entrada com as demais.

São retornados os índices das notícias mais similares, ou seja, as que possuem uma pontuação mais próxima à 1, pois quanto maior for a ponderação, maior será a probabilidade de estarem inseridas no mesmo contexto. Por fim, é desenvolvida uma ferramenta, possibilitando ao leitor que se encontra consumindo conteúdo em qualquer um dos três websites, serem notificados com links de notícias dos outros portais, sendo possível direcioná-lo para conteúdos com temas e assuntos em comum.

3.4. Desenvolvimento da ferramenta

Para o desenvolvimento da ferramenta, foi criado inicialmente uma API [5] utilizando o framework Flask[6]. O objetivo é a criação de um *endpoint* onde é possível obter as notícias mais similares relacionadas à uma notícia de entrada. Para fazer a consulta de similaridade é necessário passar como parâmetro para a API, a URL da notícia de entrada. Com isto, a API coleta as informações da notícia como o corpo e a data. Após a obtenção dos dados é realizado um filtro das notícias para que sejam analisadas apenas as que estejam no intervalo de 7 dias em relação à data da notícia de entrada. Após essa fase, o texto é processado usando o *bag-of-words* e TF-IDF para que seja aplicado o algoritmo LSI. Por fim é utilizado o cálculo da *Similaridade de Cosseno*, retornando as mais semelhantes à notícia pesquisada.

É possível analisar na figura 2, o fluxo completo da ferramenta. A extensão que encontra-se no navegador, envia uma solicitação ao servidor, no qual passa por todo o processo de cálculo de similaridade, consultando notícias com proximidade de data no banco de dados. Em seguida as notícias mais semelhantes são retornadas pelo servidor para à extensão do navegador. Ocorre em paralelo a inserção de novas notícias através de scripts automatizados que executam os spiders.

[5] https://github.com/joaonetoo/plugin
[6] http://flask.pocoo.org/

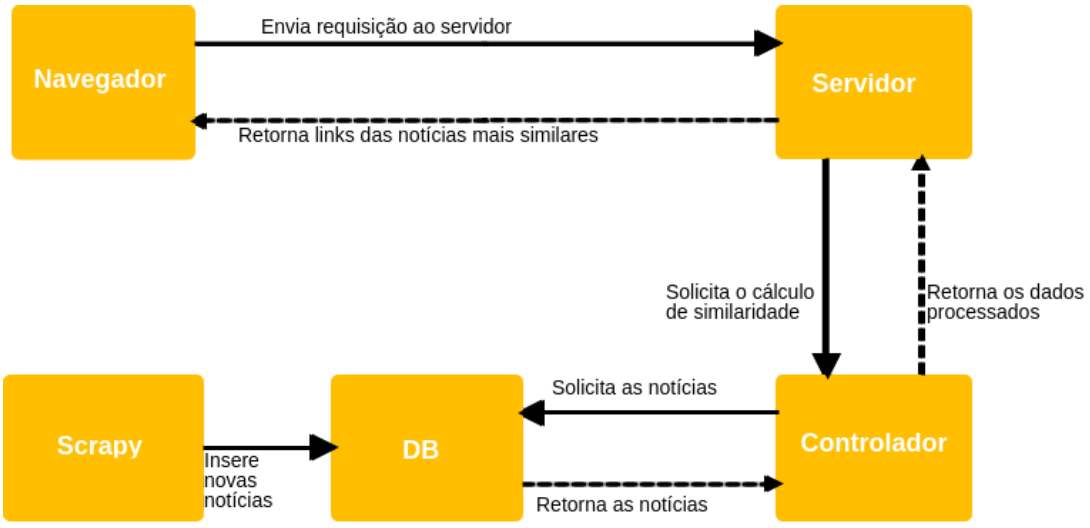

Figura 2. Fluxo da ferramenta desenvolvida

Para prover uma interface de fácil acesso para os usuários, foi desenvolvida uma extensão do Google Chrome, que lista as notícias similares a notícia da aba ativa do navegador.

Na figura 3 é possível observar o comportamento da ferramenta. Para utilizá-la é necessário abrir o *popup*, onde aparecerá um modal com um botão de "obter notícias'. Ao clicar no botão, será feita uma requisição à API, onde o resultado retornado por ela é listado na ferramenta.

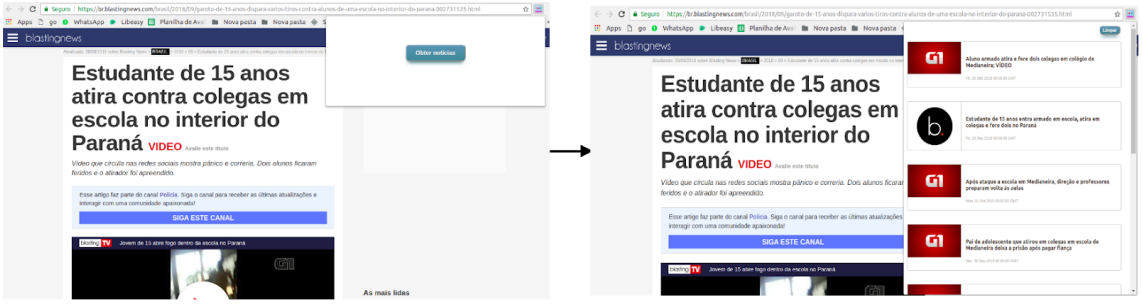

Figura 3. Funcionamento da Extensão do Chrome.

4. Análise dos resultados

Com o objetivo de analisar a similaridade de notícias classificadas pelo algoritmo aplicando a técnica LSI, foi elaborado um questionário para que fossem avaliadas as sugestões feitas pelo algoritmo para 10 notícias. Para cada notícia foram sugeridas três outras notícias similares pelo algoritmo. Para elaboração do questionário foi utilizada a escala de Likert, com o propósito de obter a opinião baseado no nível de concordância com as sugestões do algoritmo. Para cada sugestão foi necessário responder se ela era "Muito similar", "Similar", "Não sei", "Diferente" ou, "Muito Diferente".

A similaridade retornada pelo algoritmo varia no intervalo de 0 até 1. Assim, a atribuição de valores às respostas das escalas também será neste intervalo. A escala que foi utilizada é simétrica, sendo possível atribuir os seguintes valores:

- *Muito similar = 1*

- *Similar = 0,75*
- *Não sei = 0,5*
- *Diferente = 0,25*
- *Muito diferente = 0*

As notícias selecionadas para o questionário foram escolhidas aleatoriamente e os valores de similaridade salvos em um tabela para posterior comparação com a opinião dos perguntados. Na tabela 1, é possível visualizar as 10 notícias que foram aplicadas no questionário.

Tabela 1. Notícias selecionadas para elaboração do questionário

#	Título da notícia
1	Estudante de 15 anos entra armado em escola, atira em colegas e fere dois no Paraná
2	Ziraldo é transferido de CTI para unidade semi-intensiva após AVC
3	Bolsonaro sinaliza que não quer privatizar setor elétrico, BB e Caixa
4	Museu Nacional deve ser reconstruído com ajuda do Ministério da Educação
5	Facebook perde mais de US$ 100 bilhões em valor de mercado
6	TCU entrega à Justiça Eleitoral lista com mais de 7,4 mil gestores com contas julgadas irregulares
7	Datafolha para presidente, votos válidos: Bolsonaro, 58%; Haddad, 42%
8	WhatsApp limita mensagens na Índia após notícias falsas levarem a linchamentos
9	Irmã de Marielle Franco diz que foi agredida verbalmente por apoiadores de Jair Bolsonaro
10	TRF-4 nega dois recursos para que Lula participe de entrevistas

O id da notícia é vinculado ao número da questão em que a notícia aparece no formulário. Assim, a notícia que possui id 1, está no enunciado da questão 1. Foram coletadas ao final, 33 respostas do questionário. Através da análise dos resultados obtidos, foi possível obter a representação da Tabela 2. A coluna 'Notícias' contém as notícias sugeridas pelo formulário, onde 1.1 significa que a é a primeira notícia recomendada para a primeira notícia do questionário, 1.2 é a segunda notícia recomendada para a primeira notícia do questionário, 2.1 é a primeira notícia recomendada para segunda notícia do formulário e assim sucessivamente.

Tabela 2. Representação do score dos usuários, similaridade pontuada pelo algoritmo e o erro quadrático obtido da relação entre o algoritmo e o score.

ID Notícias	Muito Similar	Similar	Não sei	Diferente	Muito Diferente	Score	Algoritmo	Erro Médio
1.1	22	8	3	0	0	0,894	0,9	0,000036
1.2	13	14	3	3	0	0,78	0,925	0,021025
1.3	22	9	0	2	0	0,886	0,858	0,000784
2.1	16	11	1	4	1	0,78	0,96	0,0324
2.2	12	11	3	7	0	0,712	0,935	0,049729
2.3	7	11	0	12	3	0,553	0,95	0,157609
3.1	1	9	4	11	8	0,379	0,646	0,071289

3.2	1	4	1	13	14	0,235	0,636	0,160801
3.3	1	13	4	8	7	0,447	0,633	0,034596
4.1	6	18	2	6	1	0,667	0,908	0,058081
4.2	5	6	2	11	9	0,402	0,888	0,236196
4.3	4	7	3	12	7	0,417	0,888	0,221841
5.1	7	14	5	7	0	0,659	0,852	0,037249
5.2	5	9	4	11	4	0,5	0,845	0,119025
5.3	12	14	4	2	1	0,758	0,83	0,005184
6.1	9	14	1	9	0	0,674	0,944	0,0729
6.2	0	13	5	10	5	0,447	0,905	0,209764
6.3	0	5	5	12	11	0,28	0,77	0,2401
7.1	3	10	1	12	7	0,424	0,85	0,181476
7.2	23	7	0	2	1	0,871	0,859	0,000144
7.3	2	15	2	9	5	0,5	0,787	0,082369
8.1	9	9	2	8	5	0,568	0,922	0,125316
8.2	12	12	3	4	2	0,712	0,888	0,030976
8.3	23	6	2	0	2	0,864	0,877	0,000169
9.1	1	6	5	15	6	0,356	0,604	0,061504
9.2	1	6	3	16	7	0,333	0,573	0,0576
9.3	1	7	4	12	9	0,341	0,567	0,051076
10.1	8	15	4	5	1	0,682	0,926	0,059536
10.2	5	16	3	4	5	0,591	0,759	0,028224
10.3	5	18	4	3	3	0,644	0,88	0,055696

Para cada notícia similar, é possível calcular o *score* final da avaliação dos usuários. Para isso é utilizada a quantidade de respostas que teve cada uma das escalas. Por exemplo, na notícia 1.1, foram coletadas os seguintes valores:

- 22 pessoas responderam ‘Muito similar’.
- 8 pessoas responderam ‘Similar’.
- 3 pessoas responderam ‘Não sei’.
- 0 pessoas responderam ‘Diferente’.
- 0 pessoas responderam ‘Muito diferente’.

O score é calculado através da soma dos produtos dos valores das escalas com a quantidade de respostas para cada escala, logo em seguida é dividido pelo número total de respostas. Para o exemplo acima, o cálculo do score é feito da seguinte maneira: ((22*1) + (8*0.75) + (3*0.5) + (0*0.25) + (0*0))/33 . Com o resultado, é possível identificar a opinião dos usuários sobre a notícia sugerida. Para a notícia 1.1, o valor resultante é de 0,894, consequentemente, os usuários consideram que a notícia recomendada está entre ‘Muito similar’ e ‘Similar’. Através do cálculo do score foi possível elaborar a opinião geral dos usuários representadas na Tabela 3.

Tabela 3. Opinião dos usuários em relação às notícias sugeridas no questionário.

ID Notícias	Muito Similar	Similar	Não sei	Diferente	Muito Diferente
1.1	■	■			
1.2	■	■			
1.3	■	■			
2.1	■	■			
2.2		■	■		
2.3		■	■		
3.1			■	■	
3.2				■	■
3.3			■	■	
4.1		■	■		
4.2			■	■	
4.3			■	■	
5.1		■	■		
5.2			■		
5.3	■	■			
6.1		■	■		
6.2			■	■	
6.3				■	■
7.1			■	■	
7.2	■	■			
7.3			■		
8.1		■	■		
8.2			■		
8.3	■	■			
9.1			■	■	
9.2			■	■	
9.3			■	■	
10.1		■	■		
10.2		■	■		
10.3		■	■		

Por fim foi elaborado um apanhado geral do grau de concordância entre os usuários e o algoritmo através de uma média de erro. Para cada notícia foi subtraído o score dos usuários pelo valor de similaridade do algoritmo, elevando ao quadrado.

Ainda utilizando como amostra a notícia 1.1, o valor obtido é de 0,000036, pois é subtraído o valor do algoritmo (0,9) com o valor do score (0,894), elevando tudo ao quadrado. Logo em seguida foi aplicada a média dos erros quadráticos, somando todos os erros e dividindo a soma pela quantidade de notícias recomendadas. O valor do erro médio quadrático obtido através da equação é 0,082. Isto representa uma boa precisão do algoritmo em relação aos dados obtidos a partir da opinião dos usuários.

5. Conclusão e Trabalhos Futuros

Esta pesquisa apresentou a proposta de desenvolvimento de uma ferramenta que sugere a partir de uma notícia, outras similares. Para isso foi desenvolvido uma API que retorne às notícias similares e uma extensão do Google Chrome que consome este serviço. Para validar a integridade do algoritmo LSI aplicado na ferramenta, foi realizada uma análise através de um formulário que tem como objetivo medir o grau de concordância dos usuários em relação às notícias sugeridas pelo algoritmo.

A partir do erro médio quadrático obtido através da análise das respostas coletadas do questionário, foi possível identificar uma taxa de erro de apenas 8,2%. Logo, o algoritmo converge com a opinião dos usuários atingindo uma acurácia de 91,80%. Então, é possível afirmar que as notícias sugeridas têm um alto grau de similaridade para a amostra utilizada na pesquisa.

Para trabalhos futuros, é necessário a avaliação geral da ferramenta através de outras perspectivas como usabilidade, tempo de resposta e segurança. Além disso, é necessário elaborar novos questionários com notícias sugeridas pelo LSI aplicando uma quantidade de dimensões diferente com o propósito de aumentar a acurácia dos resultados obtidos. Para isso, é necessário elaborar um questionário com um número maior de questões e uma quantidade maior de notícias sugeridas pois em uma quantidade baixa de notícias sugeridas, é possível que as pessoas não tenham um referencial do que é uma notícia diferente, ou, muito diferente.

Referências

ALMEIDA, Raquel de Q. Fake news: arma potente na batalha de narrativas das eleições 2018. Ciência e Cultura, v. 70, n. 2, p. 9-12, 2018.

ARAÚJO, Hugo Rafael Teixeira Soares. Exploring biomedical literature using latent semantic indexing. 2012. Dissertação de Mestrado. Universidade de Aveiro.

BALDAN, Maikson A. Um Ambiente para Construção de Perfis a Partir de Textos Pessoais. 2012. Dissertação de Mestrado. Universidade Federal do Espírito Santo.

COUTO, Paloma Rodrigues Destro; LEAL, Paulo Roberto Figueira. Enquadramentos e silenciamentos: as mortes de Hugo Chávez e Margaret Thatcher pelos portais UOL e Terra. Anuário Unesco/Metodista de Comunicação Regional, v. 17, n. 17, p. 109-123, 2013.

DEERWESTER, Scott et al. Indexing by latent semantic analysis. Journal of the American society for information science, v. 41, n. 6, p. 391-407, 1990.

GEORGE, K.; JOSEPH, Shibily. Text classification by augmenting bag of words (bow) representation with co-occurrence feature. IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p-ISSN: 2278-8727, Volume, v. 16, p. 34-38, 2014.

GONZALEZ, Marco; LIMA, Vera Lúcia Strube. Recuperação de informação e processamento da linguagem natural. In: XXIII Congresso da Sociedade Brasileira de Computaçao. 2003. p. 347-395.

LUKINS, Stacy K.; KRAFT, Nicholas A.; ETZKORN, Letha H. Source code retrieval for bug localization using latent dirichlet allocation. In: 2008 15th Working Conference on Reverse Engineering. IEEE, 2008. p. 155-164.

MATSUBARA, Edson Takashi; MARTINS, Claudia Aparecida; MONARD, Maria Carolina. Pretext: Uma ferramenta para pré-processamento de textos utilizando a abordagem bag-of-words. Techinical Report, v. 209, p. 4, 2003.

PAPADIMITRIOU, Christos H., Raghavan, P., Tamaki, H., & Vempala, S. Latent semantic indexing: A probabilistic analysis. Journal of Computer and System Sciences, v. 61, n. 2, p. 217-235, 2000.

RAMOS, Juan et al. Using tf-idf to determine word relevance in document queries. In: Proceedings of the first instructional conference on machine learning. 2003. p. 133-142.

SOLKA, J. L. Text data mining: Theory and methods, Statistics Surveys, 2: 94–112. 2008.

SILVA, Allysson Costa. A influência dos parâmetros de Análise por Semântica Latente aplicada a localização de defeitos de software. 2011.

VOSOUGHI, Soroush; ROY, Deb; ARAL, Sinan. The spread of true and false news online. Science, v. 359, n. 6380, p. 1146-1151, 2018.

WIEMER-HASTINGS, Peter; WIEMER-HASTINGS, K.; GRAESSER, A. Latent semantic analysis. In: Proceedings of the 16th international joint conference on Artificial intelligence. 2004. p. 1-14.

Anexo I - Questionário

Avaliação de um algoritmo que calcula a similaridade entre notícias de websites brasileiros

O objetivo desse formulário é analisar a similaridade das notícias classificadas por um algoritmo desenvolvido no Trabalho de Conclusão de Curso do Aluno João Antonio Leite dos Santos Neto.
No formulário, estamos avaliando 10 notícias. Para cada uma delas foram sugeridas três outras notícias similares pelo o algoritmo. Com isso, você deve responder, para cada sugestão, se ela é "Muito similar", "Similar", "Não sei", "Diferente" ou "Muito diferente".

OK

* 1. Estudante de 15 anos entra armado em escola, atira em colegas e fere dois no Paraná(link)

	Muito similar	Similar	Não sei	Diferente	Muito diferente
Estudante de 15 anos atira contra colegas em escola no interior do Paraná (link)	○	○	○	○	○
Aluno armado atira e fere dois colegas em colégio de Medianeira (link)	○	○	○	○	○
Estudante atira e fere 2 alunos em escola do interior do Paraná (link)	○	○	○	○	○

* 2. Ziraldo é transferido de CTI para unidade semi-intensiva após AVC(link)

	Muito similar	Similar	Não sei	Diferente	Muito diferente
Ziraldo é internado em estado grave no Rio após sofrer AVC (link)	○	○	○	○	○
Cartunista Ziraldo sofre AVC e seu estado de saúde é grave (link)	○	○	○	○	○
Estado de Ziraldo é estável; cartunista segue no CTI (link)	○	○	○	○	○

* 3. Bolsonaro sinaliza que não quer privatizar setor elétrico, BB e Caixa (link)

	Muito similar	Similar	Não sei	Diferente	Muito diferente
Questões econômicas serão prioridade, diz Eduardo Bolsonaro. (link)	○	○	○	○	○
Questionado sobre aceno aos eleitores de centro, Bolsonaro diz que não pode virar 'Jairzinho paz e amor' (link)	○	○	○	○	○
Bolsonaro: estatais que dão prejuízo ou são cabide de emprego serão extintas (link)	○	○	○	○	○

* 4. Museu Nacional deve ser reconstruído com ajuda do Ministério da Educação (link)

	Muito similar	Similar	Não sei	Diferente	Muito diferente
Diretor do Museu Nacional e reitor da UFRJ cobram investimento federal para reconstituir local após incêndio (link)	○	○	○	○	○
Museu Nacional: os alertas ignorados que anunciavam tragédia (link)	○	○	○	○	○
Museu Nacional: os alertas ignorados que anunciavam tragédia. (link)	○	○	○	○	○

* 5. Facebook perde mais de US$ 100 bilhões em valor de mercado (link)

	Muito similar	Similar	Não sei	Diferente	Muito diferente
Zuckerberg perde US$16 bi com queda recorde de ações do Facebook. (link)	○	○	○	○	○
Um dia após queda histórica do Facebook, Twitter sofre com desconfiança do mercado (link)	○	○	○	○	○
Facebook tem maior perda diária em valor de mercado da história dos EUA (link)	○	○	○	○	○

* 6. TCU entrega à Justiça Eleitoral lista com mais de 7,4 mil gestores com contas julgadas irregulares (link)

	Muito similar	Similar	Não sei	Diferente	Muito diferente
TCU lança plataforma digital com lista de políticos que tiveram contas rejeitadas. (link)	○	○	○	○	○
Segundo Fux, candidatos ficha-suja estarão fora do 'jogo democrático (link)	○	○	○	○	○
Ministro Luiz Fux afirma em decisão que Lula é inelegível (link)	○	○	○	○	○

* 7. Datafolha para presidente, votos válidos: Bolsonaro, 58%; Haddad, 42% (link)

	Muito similar	Similar	Não sei	Diferente	Muito diferente
Datafolha mostra em quem eleitores de Ciro, Alckmin, Amoêdo e Marina declaram voto no 2º turno (link)	○	○	○	○	○
Primeira pesquisa Datafolha do 2° turno mostra Bolsonaro com 58% e Haddad com 42% (link)	○	○	○	○	○
Datafolha aponta empate técnico de Bolsonaro e Haddad entre mulheres (link)	○	○	○	○	○

* 8. WhatsApp limita mensagens na Índia após notícias falsas levarem a linchamentos. (link)

	Muito similar	Similar	Não sei	Diferente	Muito diferente
Whatsapp anuncia teste para limitar mensagens encaminhadas a várias pessoas (link)	○	○	○	○	○
WhatsApp limita o número de mensagens encaminhadas na luta contra as fake news (link)	○	○	○	○	○
WhatsApp restringe envio de mensagens após onda de linchamentos na Índia (link)	○	○	○	○	○

* 9. Irmã de Marielle Franco diz que foi agredida verbalmente por apoiadores de Jair Bolsonaro (link)

	Muito similar	Similar	Não sei	Diferente	Muito diferente
Freixo pede que polícia ouça candidato que destruiu homenagem a Marielle (link)	○	○	○	○	○
Candidato que destruiu placa de Marielle é eleito deputado no Rio (link)	○	○	○	○	○
Após ataques, campanha por placas de Marielle já arrecada R$ 28 mil (link)	○	○	○	○	○

* 10. TRF-4 nega dois recursos para que Lula participe de entrevistas (link)

	Muito similar	Similar	Não sei	Diferente	Muito diferente
Lula pede ao Supremo autorização para dar entrevistas na prisão da Lava Jato (link)	○	○	○	○	○
Dallagnol contesta decisão de Lewandowski sobre entrevista de Lula (link)	○	○	○	○	○
Deputados do PT vão ao STF para tentar permitir entrevistas de Lula. (link)	○	○	○	○	○