http://journal.unoeste.br/index.php/ce/index
DOI: 10.5747/ce.2024.v16.e423
ISSN on-line 2178-8332

Colloquium Exactarum

Submetido: 17/07/2023 Revisado: 08/05/2024 Aceito: 10/06/2024

PROCESSAMENTO DE LINGUAGEM NATURAL PARA RECONHECIMENTO DE ENTIDADES NOMEADAS EXTRAÍDAS DE CORPUS

Natural Language Processing for Recognition of Named Entities Extracted from the Corpus

Miriam Regina Bordinhon, Pedro Henrique Ignácio Leite

Centro Universitário de Adamantina, FAI, Adamantina, SP.
e-mail: miriam@fai.com.br; 90722@fai.com.br;

RESUMO – Este artigo apresenta uma implementação de código para automatizar a leitura, análise e geração de resumos de textos não estruturados da web, destacando o papel crucial do Processamento da Linguagem Natural (PLN). O processo foi dividido em duas etapas: coleta de dados e pré-processamento. Na primeira etapa, um artigo científico foi selecionado e dados relevantes foram extraídos via *web scraping*, organizados em uma página HTML hospedada online. No pré-processamento, realizaram-se tokenização, normalização, remoção de *stopwords* e contagem de palavras. Utilizando o NLTK, as sentenças mais importantes foram identificadas e ranqueadas com base na frequência das palavras-chave, permitindo a seleção das sentenças mais relevantes das seções de introdução, metodologia, resultados/discussões e conclusão. Discrepâncias como *links* e descrições de figuras foram removidas para aprimorar a clareza do resumo. O método "*most_common*()" foi utilizado para selecionar as palavras mais relevantes em cada seção. Após processamento adicional para excluir palavras desnecessárias, o resumo final mostrou-se claro e compreensível, apesar de pequenas falhas. Esta abordagem automatizada proporciona uma forma eficiente de sintetizar informações de artigos científicos, otimizando o processo de análise textual.
Palavras-chave: Processamento de Linguagem Natural; Automação; NLTK; Sumarização.

ABSTRACT – This article presents a code implementation to automate the reading, analysis and generation of summaries of unstructured web texts, highlighting the crucial role of Natural Language Processing (NLP). The process was divided into two stages: data collection and pre-processing. In the first stage, a scientific article was selected and relevant data was extracted via web scraping, organized into an HTML page hosted online. In pre-processing, tokenization, normalization, stopword removal and word counting were performed. Using NLTK, the most important sentences were identified and ranked based on keyword frequency, allowing the selection of the most relevant sentences from the introduction, methodology, results/dicussion and conclusion sections. Discrepancies such as links and figure descriptions have been removed to improve the clarity of the abstract. The “most_common()” method was used to select the most relevant words in each section. After additional processing to delete unnecessary words, the final summary was clear and understandable, despite minor flaws. This automated approach provides an efficient way to synthesize information from scientific articles, optimizing the textual analysis process.

Keywords: Natural Language Processing; Automation; NLTK; Summarization.

1. INTRODUÇÃO

O PLN, do inglês *Natural Language Processing* (NLP), é uma subárea da Ciência da Computação (Allahyari *et al.*, 2017), que surgiu em meados da década de 1950 com a publicação do trabalho *Computing machinery and intelligence* ("Máquinas de Computação e Inteligência") por Alan Turing. Na sequência, vários trabalhos foram apresentados, como a tradução automática de mais de sessenta frases em russo para o inglês, assim como sistemas estatísticos de tradução (Martins *et al.*, 2020, p.17).

Já por volta de 1970 e 1980, diversos programadores colaboraram com o que se chamou de "ontologias conceituais", que se referem à facilitação da tradução de textos médicos, e até mesmo dicionários médicos, como: MARGIE, TaleSpin, Qualm, Sam, Pam, Politics, Plot Units (Martins *et al.*, 2020, p. 17).

O primeiro software de simulação de diálogos foi o Eliza, que é um tipo de chatterbot, ou seja, programas que simulam humanos na conversação interpessoal. Já no final da década de 1980, houve a introdução de algoritmos de aprendizagem automática, *Machine Learning*. Ademais, os processos como a marcação de partes da fala (*part-of-speech tagging*), ao qual usavam algoritmos com modelos estatísticos para reconhecimento de fala, impulsionaram significativamente o avanço no campo do Processamento de Linguagem Natural. Atualmente, têm-se algoritmos de árvores de decisão, principalmente na aprendizagem de máquinas estatísticas dentro da própria aprendizagem automática. Nos últimos sete anos, destacam-se os algoritmos de aprendizagem profunda (*deep learning*) (Martins *et al.*, 2020, p. 17), ao qual tem se destacado os algoritmos de Redes Neurais Artificiais (RNA).

O grande volume de dados no meio digital se mantém em crescimento exponencial, sendo que a maioria das informações encontradas na internet está na forma não estruturada ou em linguagem natural. Os dados não estruturados são os que não possuem uma estrutura definida, como documentos textuais, e-mails, vídeos, redes sociais, bibliotecas digitais (Allahyari *et al.*, 2017). A linguagem natural usada por seres humanos para se comunicarem, se encontra no formato não estruturado. Allahyari *et al.* (2017), enfatiza que a extração de informação é a tarefa de automaticamente extrair informação estruturada a partir de textos não estruturados. Sendo assim, para extrair informação de grandes bases textuais, têm-se as técnicas de aprendizagem de máquina aplicadas a coleções de diferentes documentos, procurando extrair padrões relevantes para organizar as informações do texto em um formato que facilite o processo de recuperação da informação (Aranha; Passos, 2006 *apud* Guimarães; Meireles; Almeida, 2019, p. 171).

Através do PLN, campo da Ciência da Computação e da Linguística, é possível analisar textos e, com o uso de modelos computacionais, treinar o algoritmo a identificar regras e padrões estabelecidos pela linguagem, como, por exemplo, em português ou outra língua. Neste artigo foi tratado em português, a identificar, substantivos, artigos, numerais, verbos.

Para concluir, a relevância deste artigo se deu pela necessidade de gerar resumos automatizados eficientes. A geração manual de resumos de artigos científicos é uma atividade que exige tempo e esforço. O problema tratado foi como automatizar esse processo, permitindo que a máquina identificasse as informações mais relevantes em um texto e as sintetizasse em um resumo completo, visto que o resumo dos artigos é exigido nas publicações e relevante para a divulgação do conhecimento.

Este artigo teve como objetivo implementar e demonstrar um código capaz de realizar a análise morfológica em textos extraídos da *web* que podem ser avaliados aplicando o pré-processamento linguístico, para serem utilizados no PLN que atuou realizando a análise do texto para garantir que a extração de suas características estruturais se tornasse compreensível pela máquina, facilitando a geração do resumo de artigos científicos escritos de uma forma geral.

Este artigo está organizado em quatro seções, sendo que na Seção 2 é descrita a fundamentação teórica, que apresenta técnicas para gerar a sumarização. Na Seção 3 é apresentada a Metodologia, com as etapas utilizadas para a coleta dos dados e pré-processamento, na Seção 4, resultados e discussões, apresenta resumos automáticos gerados pelo algoritmo e quais foram as melhorias desenvolvidas para alcançar resultados significativos. Por fim, na Seção 5, são apresentadas algumas conclusões obtidas a partir das investigações realizadas, as limitações observadas e inclusive, algumas linhas de trabalhos futuros que podem ser seguidas.

2. FUNDAMENTAÇÃO TEÓRICA

2.1 Processamento da linguagem natural

O PLN é uma vertente da inteligência artificial que se dedica a capacitar os computadores a compreender, interpretar e gerar linguagem humana de maneira eficaz. Esse campo multidisciplinar combina conhecimentos da linguística, ciência da computação e aprendizado de máquina para criar sistemas capazes de interagir com as pessoas de forma natural e entender textos como humanos (Martins *et al.*, 2020).

Um dos maiores problemas relacionados ao PLN é a questão da dimensionalidade do vocabulário. A comunicação humana é muito vaga às vezes, pois a maioria das pessoas acaba utilizando coloquialismos, abreviações e muitas vezes não há preocupação em corrigir erros ortográficos, principalmente no universo da internet e dos dados digitais, que se expande não só para incluir um número cada vez maior de pessoas on-line, mas também para incluir todas as informações esperadas da Internet das Coisas (Rodríguez; Bezerra, 2020, p. 67-77).

Uma das tarefas cruciais do PLN é a análise morfológica, na qual as palavras são desmembradas em suas raízes e afixos para entender sua estrutura e significado, ajudando a identificar formas verbais, nominais, tempos verbais e outras informações linguísticas vitais para a interpretação precisa do texto. Outro aspecto importante é a análise sintática, que lida com a estrutura gramatical das frases, identificando os relacionamentos entre as palavras e como elas se combinam para formar proposições e significados. A partir disso, é possível entender a hierarquia das partes da frase e a função de cada elemento na sentença (Pinto, 2015; Barbosa *et al.*, 2017).

A classificação de entidades nomeadas também é uma aplicação valiosa do PLN, na qual o objetivo é identificar e categorizar entidades específicas mencionadas no texto, como nomes de pessoas, organizações, locais, datas e valores. Essa tarefa é crucial para extrair informações relevantes de documentos extensos e para aprimorar a compreensão do contexto em que as entidades estão inseridas (Barbosa *et al.*, 2017). Além disso, a análise semântica é um pilar fundamental do PLN, pois visa compreender o significado das palavras e como elas se relacionam uma com as outras (Pinto, 2015). Em termos gerais, a evidência primária para a linguística semântica vem das interpretações do orador nativo no uso das expressões no contexto, padrões de uso, colocação e frequência, que são detectáveis (Barbosa *et al.*, 2017).

O PLN tem uma ampla gama de outras utilidades, incluindo a geração automática de resumos de textos, a criação de *chatbots* inteligentes para interações online, a análise de opiniões e críticas de produtos, a categorização de documentos, a indexação e recuperação eficientes de informações, entre muitas outras (Martins *et al.*, 2020).

Em resumo, o PLN é uma área dinâmica e em constante evolução que desempenha um papel vital na transformação digital atual. Com seu contínuo aprimoramento e a integração com outras tecnologias, o PLN promete revolucionar a forma como se interage com a informação e como as máquinas auxiliam no processamento e compreensão da vasta quantidade de dados disponíveis (Martins *et al.*, 2020).

2.2 Sumarização

Segundo Mani (2001, apud Pardo, 2008) Pardo (2008 *apud* Mani, 2001), a sumarização de textos é uma prática valiosa e recorrente que desempenha um papel fundamental na tomada de decisões, sendo definida como a versão mais curta e condensada de um texto-fonte. No contexto brasileiro, a Associação Brasileira de Normas Técnicas (ABNT) adota o termo "sumário" para se referir a um índice, abordando a enumeração das partes de um trabalho.

O processo de sumarizar textos se tornou indispensável para a simplificação da informação, facilitando a acessibilidade, especialmente em dispositivos com telas pequenas, e na apresentação de dados para leitores com pouco domínio da língua (Tosta; Felippo; Pardo, 2012).

Devido à grande quantidade de material textual gerado diariamente na *web*, e do avanço da PLN, é de grande interesse a automatização desse processo de sumarização, foco da subárea do PLN denominada Sumarização Automática (SA) (Tosta; Felippo; Pardo, 2012).

Uma das dificuldades de se realizar o processo de SA é encontrar um padrão a se seguir, já que cada texto tem seu gênero e estrutura definida, e mesmo em texto de mesmo estilo, ainda tem um fator variável, que são as características de escrita do autor. Por essas razões, se faz necessário que se definam os conceitos relativos ao que se consideram sumários. Primeiramente, os sumários remetem

necessariamente a eventos ou textos originários do mesmo, e em segundo lugar, devem ser escritos de forma a não perder seu significado original, mesmo que ocorra a perda de algumas informações ou apresentem uma estrutura diferente da sua fonte (Martins *et al.*, 2013).

De acordo com a função, os sumários podem ser informativos, indicativos ou críticos. Quanto à forma, os sistemas de SA podem produzir extratos ou abstracts. Os extratos são sumários compostos por trechos inalterados do texto-fonte, forma utilizada neste estudo, enquanto os abstracts apresentam partes reescritas do texto-fonte (Pardo, 2008).

3. MÉTODO

Esta seção visa detalhar os processos e técnicas pela qual o código foi concebido e implementado, desde a escolha da linguagem da programação até o passo a passo do processamento, cujo propósito foi gerar resumos científicos coerentes e entendíveis ao leitor de artigos.

Para o desenvolvimento do código, foi utilizada a linguagem *Python*, pois a mesma possui diversas bibliotecas necessária para a geração automática do resumo, como as bibliotecas B*eatifulsoap* para extração dos dados e a *Natural Language Toolkit (*NLTK) para processamento dos dados, tendo esta um grande volume de recursos e funções úteis para análise dos textos, como: classificação, tokenização, *stemming*, *tagging*, *parsing* e raciocínio semântico. A biblioteca NLTK foi escolhida pela alta curva de aprendizagem, sua sintaxe transparente e pela facilidade de manipular as funções. Recomenda-se instalar alguns pacotes adicionais do NLTK, os quais fornecem um maior suporte para a língua portuguesa e podem ser selecionados na instalação do NLTK dentro do ambiente de execução. Esses pacotes incluem: floresta*,* mac_morpho*,* machado*, punkt* e *stopwords*.

Para o desenvolvimento dessa aplicação, foi necessário dividi-la em dois módulos distintos: coleta de dados e pré-processamento. Na etapa de coleta dos dados e escolha do material textual, selecionou-se um artigo científico publicado, "Avaliação das etapas de pré-processamento e de treinamento em algoritmos de classificação de textos no contexto da recuperação da informação" (Guimarães; Meireles; Almeida, 2019). Já para a fase de pré-processamento, utilizou-se a técnica de *web scraping*, que pode ser definida como o processo de extração e combinação de conteúdos de interesse da *web* de forma sistemática. Nesse processo, um agente de software, também conhecido como robô *web*, imita a interação de navegação entre os servidores *web* e o ser humano em uma travessia convencional da Web. Passo a passo, o robô acessa quantos sites forem necessários, analisa seus conteúdos para encontrar e extrair dados de interesse e estrutura esses conteúdos conforme desejado (Glez-Peña *et al.*, 2014). Para isso, criou-se uma página HTML contendo o conteúdo extraído do referido artigo científico. Em seguida, hospedou-se essa página em um site gratuito, dando início à primeira etapa do pré-processamento textual: a *tokenização*. A *tokenização* é uma operação utilizada pelo mecanismo de análise de texto, que segmenta o texto em unidades chamadas *tokens*, para assim, conduzir a **análise morfológica**, podendo ser delimitadas por pontos finais (.), vírgulas (,), ponto e vírgulas (;) e outros caracteres. Durante este processo, pontuações e caracteres especiais são completamente removidos. O próximo passo consiste na **análise lexical**, que relaciona as variantes morfológicas das palavras aos seus *lemmas*, também conhecida como análise de *parsing side*, que nada mais são do que as formas canônicas das palavras. Dando sequência ao processamento realizado, tem-se a **análise semântica,** que se refere à análise do significado das palavras, expressões fixadas, sentenças e enunciados no contexto. Em termos gerais, vem das interpretações das expressões, padrões de uso, colocação e frequência, que são detectáveis usando técnicas linguísticas em córpus. Por fim, a **análise pragmática**, que busca interpretar a mensagem, extraindo informações e significados implícitos nas palavras (Dale, 2010 *apud* Faria; Barbosa, 2020, p. 1376).

Para a análise em PLN, foram seguidas as etapas descritas na Figura 1, o processo iniciou com a entrada do texto, seguido pela *tokenização* e todas as etapas de análise tradicionais, culminando na obtenção do significado desejado pelo analisador. Durante essas etapas, o texto foi dividido em unidades lógicas, como frases e *tokens,* que foram utilizadas para construir a estrutura gramatical do texto. Os *tokens* de texto podem passar por normalização por meio de *stemming* ou *lematização* (Kononova *et al.*, 2021, p. 4). Além disso, as *stopwords* foram utilizadas para remover palavras que ocorrem frequentemente, como artigos, preposições, pontuações, conjunções e pronomes. Por fim, realizou-se a contagem de palavras, uma etapa essencial para identificar a frequência de cada palavra, sendo esse um elemento-chave na identificação das frases que comporão a sumarização.

Figura 1. Etapas de análise em PLN

Fonte: Os autores.

Foi preciso criar e hospedar uma página *web* contendo o conteúdo textual do artigo selecionado para a realização da extração de dados. Esta página foi hospedada através de um serviço online chamado *netlify* que oferece hospedagem gratuita.

Para acessar a página *web* hospedada e realizar o processamento do texto, foi necessário importar as funções '*Request'* e '*Urlopen'* da biblioteca '*urllib.request*'. O módulo '*Urllib*' é utilizado para recuperar *Uniform Resource Locators* (URL) e '*urllib.request'* é empregado para abrir e ler os módulos necessários para trabalhar com a URL. Essas funções foram essenciais para ler a página *web* criada e hospedada, sendo necessário armazenar todo o conteúdo em uma variável.

Para realizar o *Web Scraping* foi importada a biblioteca *Beautiful Soup*, sendo esta apresentada na versão quatro.

Ao analisar o código-fonte da página, nota-se que o conteúdo textual estava contido em diferentes elementos "div" com identificadores distintos, como "Introdução", "Metodologia", "Resultado" e "Conclusão". Esses identificadores foram usados para agrupar os elementos para fins de estilo. No caso, o comando "*find*()" foi utilizado para extrair o conteúdo dessas variáveis, obtendo apenas o texto relevante. O conteúdo extraído foi armazenado em quatro variáveis, utilizando o método "*getText*()". Esse procedimento foi necessário para selecionar apenas as partes específicas do texto do artigo que seriam utilizadas para gerar o resumo (Introdução, Metodologia, Resultado e Conclusão).

Na próxima etapa, o PLN é aplicado ao texto extraído. Para isso, foram importadas algumas funcionalidades do NLTK responsáveis pela "tokenização" dos textos. As funções utilizadas foram '*word_tokenize'* e '*send_tokenize'*.

Após realizadas as importações, aplicou-se a tokenização, tal procedimento fez a normalização do texto. Para isso, o texto foi dividido em sentenças, através do comando "*sent_tokenize*()", sendo elas salvas em variáveis distintas de acordo com os tópicos do texto, e posteriormente em palavras, além disso, por meio do comando "*word_tokenize*()" e do uso da função "*lower()*", as palavras permaneceram mapeadas para versões em letras em minúscula. Essa etapa de pré-processamento ajuda a evitar duplicações desnecessárias de palavras.

O *word_tokenize*() utiliza o parâmetro '*language'* para realizar a tokenização interna das sentenças utilizando o '*PunktSentenceTokenizer'* e, em seguida, realiza a tokenização das palavras de cada sentença utilizando o '*NLTKWordTokenizer'*.

O processo de "*tokenização*" do NLTK considera as pontuações do texto como *tokens*. Portanto, é necessário removê-las juntamente com as chamadas "*stopwords*", que podem interferir nos processos realizados. Para fazer isso, importam-se as bibliotecas "*stopwords*" e "*punctuation*". Em seguida, por meio de uma estrutura de repetição, realiza-se a remoção dessas "impurezas" presentes nas variáveis criadas para este armazenamento específico.

O próximo passo consiste em criar uma distribuição de frequência para as listas de palavras criadas anteriormente. Para identificar as palavras mais relevantes, utilizou-se a função "*FreqDist*" importada da biblioteca "nltk.probability".

Após a criação da distribuição de frequência, é necessário identificar quais são as sentenças mais importantes do texto. Para isso, atribui-se um "*score*" a cada sentença com base no número de ocorrências da palavra-chave relevante dentro delas. Para realizar esse processo, utiliza-se um dicionário chamado "*defaultdict*" da biblioteca "*collections*".

Ao aplicar o processo descrito, foi possível identificar as sentenças mais relevantes. Em seguida, foi gerada uma estrutura de repetição que percorreu todas as sentenças, somando a frequência das palavras nelas contidas. Após a identificação das sentenças mais importantes, foi realizada a seleção dessas sentenças utilizando a funcionalidade "*nlargest*" da biblioteca "*heapq*". Desta maneira, para gerar a sumarização do artigo, foram extraídas três sentenças da Introdução, três sentenças da Metodologia e uma única sentença de cada para os Resultados e Conclusão. No total, oito sentenças são combinadas para a impressão do resumo. É importante ressaltar que a quantidade de sentenças pode ser livremente alterada pelo usuário, o que permite gerar resumos diferentes. Todo o processo de impressão do resumo é baseado no método descrito.

4. RESULTADOS E DISCUSSÕES

Inicialmente, ao gerar os primeiros resumos, percebeu-se a necessidade de dividir o texto em seções específicas. Isso ocorre porque um resumo de um artigo, como o utilizado na pesquisa, requer dados da introdução, metodologia, resultados e discussões, e, por fim, a conclusão, sem essa segmentação o resumo gerado acabava por não incluir trechos importantes para o entendimento do resumo. Portanto, foram extraídas partes desses textos para realizar as etapas de análise em PLN. Essa etapa foi crucial, pois caso não fosse realizada, informações desnecessárias, como o título do artigo, links de imagens, tabelas contidas no texto e citações de autoria presentes nas referências bibliográficas, poderiam ser extraídas.

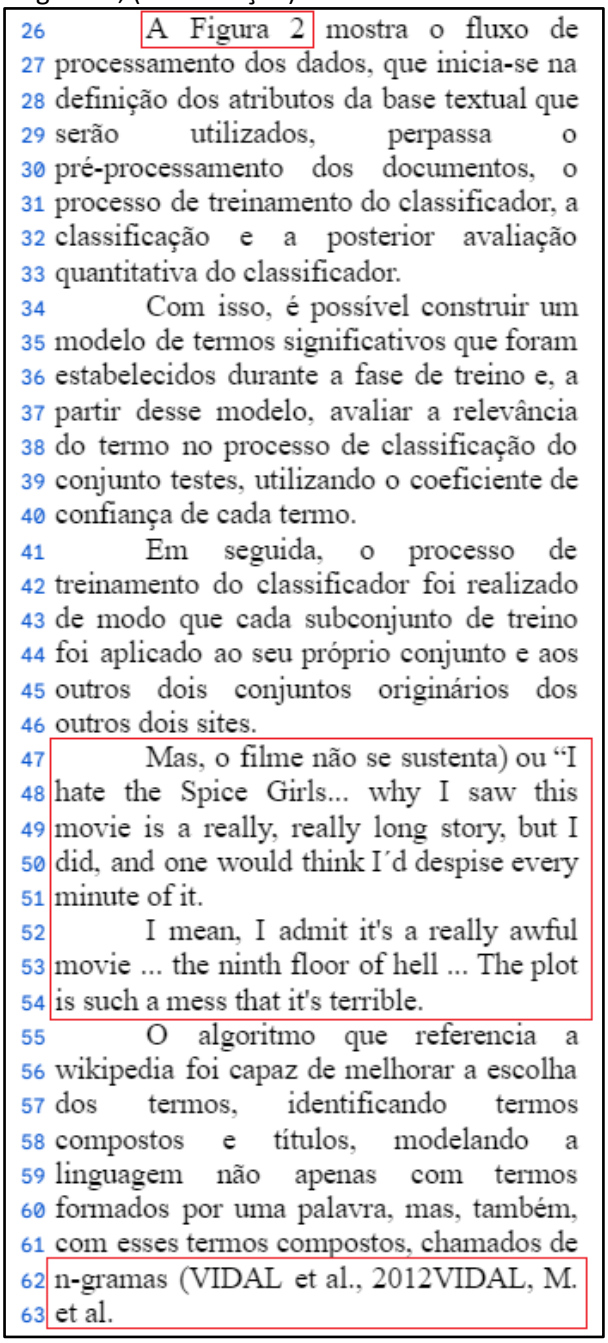
Os dados obtidos para a aplicação dos testes foram coletados a partir do desenvolvimento da página *web*. Entretanto, ao realizar uma sumarização completa do texto, foi identificado que o resumo apresenta discrepâncias em sua forma de apresentação, demonstrado nas Figuras 2 e 3.

Figura 2. Primeira sumarização gerada.

2018.https://arxiv.org/pdf/1707.02919.pdf..
), extração de informação é a tarefa de automaticamente extrair informação estruturada a partir de textos não estruturados, que constituem a forma mais simples de dados gerada em diferentes contextos.
A classificação automática de documentos de texto utiliza diferentes técnicas de aprendizado de máquina aplicadas a coleções diferentes de documentos, procurando extrair padrões relevantes para organizar as informações do texto em um formato que facilite o processo de RI (ARANHA; PASSOS, 2006ARANHA, C.; PASSOS, E. A Tecnologia de Mineração de Textos.
Uma das etapas fundamentais do processo de classificação de textos é o pré-processamento, na qual um conjunto de ações transformadoras é aplicado, para garantir que a informação em formato não estruturado (texto) se torne passível de compreensão pela máquina.

Fonte: Os autores.

Figura 3. Primeira sumarização gerada, (continuação).



Fonte: Os autores.

Dentre as divergências apresentadas, destaca-se o *link*: "2018.https://arxiv.org/pdf/1707.02919.pdf" no início do resumo, demonstrado em vermelho na Figura 2, linhas 1 e 2, ao qual não foi selecionado, porém, atribuiu como parte do resumo inicial.

Outra incoerência se destaca, em vermelho na Figura 2, linhas 16 a 18, como a apresentação de algumas citações de autoria como em: "(Aranha; Passos, 2006; Aranha, C.; Passos, E. A Tecnologia de Mineração de Textos", assim como também nas linhas 62 e 63 da Figura 3, sendo estas também desnecessárias em uma sumarização.

Outro fato, como se destaca na descrição de "figura 2", desnecessária na geração de resumos, evidenciado na Figura 3, linha 26, observando também algumas partes impressas em outra língua, linhas 47 a 54, no caso em inglês.

Após uma revisão sobre a estrutura do código verificou que estas discrepâncias decorriam da escolha das sentenças, que por sua vez continham esses erros, então foi adicionada mais uma etapa de processamento para remover estas palavras indesejadas do texto, o que não apresentou melhorias, a solução encontrada foi remover as sentenças das etapas de processamento subsequentes.

No código desenvolvido, realizou-se a contagem da frequência das palavras em cada seção do texto (Introdução, Metodologia, Resultados/Discussões e Conclusão). Presumivelmente, em cada uma dessas seções, foi criada uma variável correspondente para cada item, nomeadas como “FreqI”, “FreqM”, “FreqR” e “FreqC”, respectivamente. O código itera sobre cada uma e usa o método “*most_common*()” para obter uma lista das palavras mais comuns e suas frequências em cada seção. Isso permitiu visualizar quais palavras ocorrem com mais frequência em cada seção específica do artigo, observar o resultado na Tabela 1, permitindo, assim, a análise das palavras de maior peso no cálculo das notas das sentenças destinadas a participar da geração da sumarização.

Tabela 1. Frequência das palavras

Seção	Palavra	Frequência
Introdução	informação	7
	processo	7
	textos	6
Metodologia	treinamento	6
	conjunto	5
	avaliação	5
Resultado	processo	8
	experimentos	7
	classificação	6
	I	6
Conclusão	termos	4
	trabalho	4

Fonte: Os autores.

Estas palavras de maior frequência não necessariamente aparecem na sumarização gerada. A função que calcula a pontuação das sentenças inclui todas as palavras, de modo que frases maiores podem ter preferência sobe uma frase menor, mesmo que contenham palavras de alta incidência. Além disso, as palavras na tabela não estão em sua forma radical, podendo aparecer escritas de maneira diferente na sumarização.

Nos parágrafos iniciais do resultado da sumarização, percebe-se a ocorrência das palavras “informação”, “processo” e “textos”, sendo estas retiradas da Introdução do artigo trabalhado. Já nos quinto, sexto e sétimo parágrafos, Figura 3, linhas 26 a 46, foram selecionadas as palavras “treinamento”, “conjunto” e “avaliação” retiradas da Metodologia.

No oitavo e nono parágrafos, Figura 3, linhas 47 a 54, apenas a palavra em inglês “I”, que por aparecer repetida vezes teve as duas sentenças da análise selecionadas. Das linhas 55 a 63 a palavra “termos” teve maior ocorrência, sendo esta selecionada para criar a Conclusão da sumarização.

Após adicionar algumas etapas de processamento que percorre o texto palavra por palavra através de uma estrutura de repetição, foi possível identificar e eliminar palavras desnecessárias para a apresentação da sumarização, como por exemplos: links, “figura”, autoria, palavras em outro idioma, conforme demonstrado anteriormente. Após esse processo, observou-se um resultado relativamente avançado, (Figura 4). Embora possa haver algumas falhas na escrita e uma menção a uma coluna de tabela, (destaque em caixa vermelha), “A segunda coluna”, ainda é possível compreender o assunto de forma geral. Como se pode verificar, houve um progresso significativo se comparado com o primeiro resumo gerado.

Figura 4. Segunda sumarização gerada.

O processamento dos textos possibilita que as avaliações sejam classificadas como positivas ou negativas, auxiliando o processo de recuperação desta informação. Esta pesquisa propõe uma avaliação quantitativa das etapas de pré-processamento e de treinamento de um classificador no processo de classificação automática de dados não estruturados. Foram utilizados três conjuntos de dados extraídos de sites que oferecem produtos ou serviços que podem ser avaliados. Foi utilizado para a análise um conjunto de avaliações de usuários submetidos aos sites Amazon, IMDB e Yelp. Em seguida, o processo de treinamento do classificador foi realizado de modo que cada subconjunto e os outros dois conjuntos originários dos outros dois sites. Para iniciar o processo de classificação, foi necessário estabelecer um conjunto de dados utilizado para o treinamento do classificador e outro para teste. Os sites escolhidos fornecem um conjunto distinto (produto, serviços e filmes) de elementos a serem avaliados. Apresentam as métricas de avaliação dos experimentos de classificação da avaliação dos sites IMDB e Yelp, respectivamente, utilizando-se para o treinamento o subconjunto da Amazon. A segunda coluna apresenta as mesma métricas para a classificação dos sentimentos negativos. O pré-processamento dos textos disponíveis possibilitou observar que, ainda que as possibilidades de avaliação em contextos distintos sejam iguais (ou seja, positiva ou negativa), o elemento a ser avaliado possui termos considerados relevantes pelo pré-processamento tradicional que não necessariamente o são. Os resultados obtidos neste trabalho, mostram que as técnicas de classificação disponíveis conseguem alcançar resultados satisfatórios. Quanto mais subjetivo, mais sarcástico ou mais irônico o teor da avaliação do usuário, mais difícil é, para a máquina, classificar o texto como sendo o de uma avaliação positiva ou negativa.

Fonte: Os autores.

5. CONSIDERAÇÕES FINAIS

Este artigo teve como objetivo realizar a sumarização de um artigo científico, com base na extração dos dados para criar uma página HTML e, posteriormente, extrair as informações necessárias para o processamento e geração do resumo. No primeiro resumo gerado, foi necessário dividir o conteúdo em quatro seções principais: “Introdução”, “Metodologia”, “Resultados” e “Conclusão”. Na sequência, fez a escolha de quantas sentenças seriam necessárias para gerar o resumo, e após vários testes, identificou-se que oito sentenças forneciam os melhores resultados para o artigo em questão. Isto não impede que em outros artigos seja diferente. No entanto, observaram-se divergências nos resultados em relação às citações de autoria, figuras e outros elementos, indicando a necessidade de ajustes, correções na técnica utilizada para, assim, obter uma sumarização mais adequada.

Uma área que pode ser aprimorada é o método de captura dos dados, de modo que não seja necessário analisar o código HTML da página para identificar as estruturas em que os textos estão localizados, mas sim apenas o *link* da página. Isso permitiria obter dados de melhor qualidade para análise, evitando a dependência do código HTML e simplificando todo o processo.

Uma forma de aprimorar a confiabilidade desses sistemas é por meio do desenvolvimento de algoritmos mais avançados e do treinamento de modelos com conjuntos de dados mais diversificados e representativos. Essas abordagens têm o potencial de aprimorar significativamente a confiabilidade dos sistemas, permitindo um processamento mais preciso e abrangente das informações.

Como trabalho futuro, pretende-se gerar todos os procedimentos de processamento utilizando a biblioteca Spacy. Essa biblioteca é conhecida por sua eficiência e desempenho, sendo frequentemente escolhida para tarefas de PLN em projetos de pesquisa e desenvolvimento. Ela fornece uma API fácil de usar e é compatível com vários idiomas, além de oferecer suporte a modelos pré-treinados para diferentes tarefas de PLN.

AGRADECIMENTOS

Agradecemos ao Centro Universitário de Adamantina por disponibilizar a bolsa ao aluno através do PROBIC/UniFAI.

REFERÊNCIAS

ALLAHYARI, M. *et al.* A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques**.** *In*: CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, 23., 2017, Halifax, **Proceedings** […]. Halifax, SIGKDD, 2017. Disponível em: https://arxiv.org/pdf/1707.02919.pdf. Acesso em: 24 mar. 2022.

BARBOSA, J. *et al.* Introdução ao Processamento de Linguagem Natural usando Python*. In*: ESCOLA REGIONAL DE INFORMÁTICA DO PIAUÍ. 3., 2017, Teresina, PI. **Anais** [...]. Teresina: [*s.n.*], 2017, v. 1, n.1, p. 336-360, 2017. Disponível em: tutorial_nltk.pdf (ufu.br). Acesso: 9 out. 2023.

BENÍCIO, D.H.P. **Aplicação de mineração de texto e processamento de linguagem natural em prontuários eletrônicos de pacientes para extração e transformação de texto em dado estruturado**. 2020. Dissertação (Mestrado) - Universidade Federal do Rio Grande do Norte, Natal, RN, 2020.

BRITO, P. F.; ARAÚJO, L. G. A. Desenvolvimento do Módulo de Pré-processamento da Ferramenta SentimentALL. **Digital Object Identifier**, v. 1, n. 1, p. 2019. DOI https://doi.org/10.33911/singular-etg.v1i1.22.

FARIA, C. R.; BARBOSA, C. R. S. C. Técnicas de Processamento de Linguagem Natural para Auxiliar o Estudante na Identificação das Pragas da Soja. *In*: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO (SBIE), 31. , 2020, Porto Alegre. **Anais** […]. Porto Alegre: Sociedade Brasileira de Computação, 2020 . p. 1373-1382. DOI: https://doi.org/10.5753/cbie.sbie.2020.1373.

GUIMARÃES, L. M. S.; MEIRELES, M. R. G.; ALMEIDA, P. E. M. Avaliação das Etapas de Pré-processamento e de Treinamento em Algoritmos de Classificação de Textos no Contexto da Recuperação da Informação. **Perspectivas em Ciência da Informação**, v. 24, n. 1, p. 169-190, 2019. DOI https://doi.org/10.1590/1981-5344/3505.

KONONOVA *et al.* Opportunities and challenges of text mining in materials research. **iScience**, v. 24, n. 3, p. 102155, 2021. DOI https://doi.org/10.1016/j.isci.2021.102155.

GLEZ-PEÑA, D. *et al.* Web scraping technologies in an API world. **Briefings in bioinformatics**, v. 15, n. 5, p. 788–797, 2014. DOI https://doi.org/10.1093/bib/bbt026.

MARTINS, J. S. *et al.* **Processamentos de Linguagem Natural**. São Paulo: Grupo A, 2020.

MARTINS, C. B. *et al.* **Introdução à Sumarização Automática**. 2013. Disponível em: https://sites.icmc.usp.br/taspardo/RTDC00201-CMartinsEtAl.pdf. Acesso em: 9 out. 2023.

PARDO, T. A. S. **Sumarização Automática**: Principais Conceitos e Sistemas para o Português Brasileiro. Série de Relatórios do Núcleo Interinstitucional de Linguística Computacional NILC-ICMC-USP, 2008.

PINTO, S. C. S. **Processamento de linguagem natural e extração de conhecimento**. 2015. Dissertação (Mestrado em Engenharia Informática) - Faculdade de Ciências e Tecnologia da Universidade de Coimbra, Coimbra, PO, 2015.

RODRÍGUEZ, M. M. M. S.; BEZERRA, B. L. D. Processamento de Linguagem Natural para Reconhecimento de Entidades Nomeadas em Textos Jurídicos de Atos Administrativos (Portarias). **Revista de Eng. e Pesquisa Aplicada**, v. 5, n. 1, (Ed. Esp.), p. 67-77, 2020. DOI https://doi.org/10.25286/repa.v5i1.1204.

TOSTA, F. E. S.; FELIPPO, A. D.; PARDO, T. A. S. **Aplicação de métodos clássicos de sumarização automática no contexto multidocumento multilíngue:** primeiras aproximações**.** Série de Relatórios do Núcleo Interinstitucional de Linguística Computacional NILC-ICMC-USP, 2012.