IRIS: Extração, Organização e Classificação de Conteúdos do Projeto Pedagógico de Curso do Técnico em Automação Industrial

Vinícius Melo Almeida
Instituto Federal da Bahia - IFBA
Salvador, Bahia - Brasil
vinimelo@riseup.net

Marcelo Cad
Instituto Federal da Bahia - IFBA
Salvador, Bahia - Brasil
marcelocad@ifba.edu.br

Vitor Filardi
Instituto Federal da Bahia - IFBA
Salvador, Bahia - Brasil
vitorleao@ifba.edu.br

ABSTRACT

Traditional search systems on the World Wide Web have revolutionized searching for information, almost wholly replacing physical encyclopedias. However, these systems must develop new search methods that make the process more efficient and comprehensive for students. In technical courses, it is common for students to struggle with searching for quality content on the internet that serves both as a bibliographic base for the disciplines and as reference material to deepen their knowledge. The Intelligent Research Interactive System (IRIS) automates searches for contents of the fields in the Industrial Automation Technical Course to contribute to an effective improvement in the quality of their studies and reduce the time spent in processes that are not linked to the education activities.

PALAVRAS-CHAVES

WWW, Web Scraping, Aprendizado de Máquina, Motor de Busca.

1 Introdução

O Curso Técnico em Automação Industrial faz parte do Eixo Tecnológico de Controle e Processos Industriais, conforme descrito no Catálogo Nacional de Cursos Técnicos [1]. Em seu processo formativo, o técnico em automação concatena as exposições realizadas em salas de aula, o conteúdo disponível no acervo bibliotecário (podendo este ser físico ou virtual), conteúdos extras disponibilizados pelos docentes, além da própria prática profissional que é exigida para a conclusão do curso.

Na realidade, os discentes do curso técnico dependem da disponibilização de recursos didáticos por parte dos docentes, como apostilas e slides, para realizar as atividades previstas no plano pedagógico do curso. Isto porque o conteúdo das disciplinas do núcleo tecnológico do Curso Técnico em Automação Industrial encontra-se disperso em diferentes páginas indexadas por mecanismos de busca convencionais da internet. Boa parte destes conteúdos estão vinculados aos setores de pesquisa e desenvolvimento de empresas públicas e privadas, ou mesmo fazem parte do acervo técnico disponibilizado aos seus discentes e funcionários.

Diante deste cenário, a ausência de uma centralidade para a consulta de conteúdos do Curso Técnico em Automação Industrial, dispersos em páginas da World Wide Web (WWW), leva a um gasto de tempo maior por parte dos discentes em processos repetitivos de buscas manuais por estes conteúdos. Sabe-se que a pouca capacidade de aprendizagem e habilidade de estudo impactam diretamente na permanência de estudantes, tornando-se vetores da evasão escolar de discentes, como demonstra uma pesquisa recente da evasão escolar dos cursos técnicos integrados ao ensino médio da Rede Federal [2].

Devido ao crescimento relevante do número de usuários, a Web tornou-se uma ferramenta indispensável de comunicação e pesquisa, ocupando hoje uma posição de destaque no sistema de ensino brasileiro, como aponta a pesquisa TIC Educação em 2019 [3]. Nas últimas duas décadas, mecanismos de busca da Web ganharam um imenso poder de processamento e diversas novas funcionalidades. Contudo, mesmo com seus significativos avanços, esses mecanismos ainda falham em entregar respostas de forma eficiente aos seus usuários.

De acordo com o artigo “Search Needs A Shake-up” [4], ao invés de apresentar longas listas de documentos que contêm as palavras buscadas, os usuários precisam de respostas diretas para os seus questionamentos. O artigo aponta que é necessário um investimento maior em estratégias que permitam buscas e respostas em linguagem natural, no lugar de oferecer um índice eletrônico similar aos encontrados no fundo de livros. Depreende-se desta análise, portanto, que os serviços de busca da internet podem confundir os discentes ao apresentar milhões de resultados em resposta a uma simples busca, e esse problema só aumenta à medida que aumentam o número de páginas da Web e a busca por conteúdos através de dispositivos móveis.

Nesse sentido, o Sistema Inteligente de Busca Interativa (IRIS) visa contribuir para a formação dos discentes do curso técnico em Automação Industrial, nas modalidades integrada, concomitante e subsequente, ao realizar buscas por conteúdos das disciplinas do núcleo tecnológico (não propedêuticas). Assim, o projeto contribui para assegurar a educação inclusiva e equitativa e, de qualidade, um dos Objetivos de Desenvolvimento Sustentável (ODS) da Agenda 2030 da Organização das Nações Unidas [5].

2 Desenvolvimento

No desenvolvimento do sistema proposto, foram implementados módulos com a finalidade de cumprir objetivos específicos,

distribuindo responsabilidades entre tarefas de extração, organização e classificação.

Em linhas gerais, o módulo de extração fica responsável por fazer as requisições e extrair não apenas os links, como também os títulos e parágrafos iniciais de cada página. Uma vez resgatadas essas informações, o módulo de organização é responsável por receber e armazenar estas informações em um banco de dados relacional, juntamente com o termo de pesquisa que levou o módulo de extração até o link em questão. Posteriormente, o módulo de classificação analisa o termo de pesquisa, título, link e os parágrafos iniciais de cada conteúdo e contrasta com alguns exemplos disponíveis em um dataset (coleção de dados) de treino, de modo a permitir uma rotulação por meio de aprendizado supervisionado.

Finalizada a etapa de classificação, o rótulo é salvo junto ao conteúdo existente no banco de dados. Por fim, esses conteúdos são disponibilizados em um sistema web online para que docentes e profissionais formados no Eixo Tecnológico de Controle e Processos Industriais possam avaliar as classificações automáticas da ferramenta de forma positiva ou negativa. Deste modo, é produzido subsídio para a construção de um dataset de treinamento maior, levando a resultados de classificação mais precisos no futuro.

A arquitetura proposta é mostrada na Figura 1.

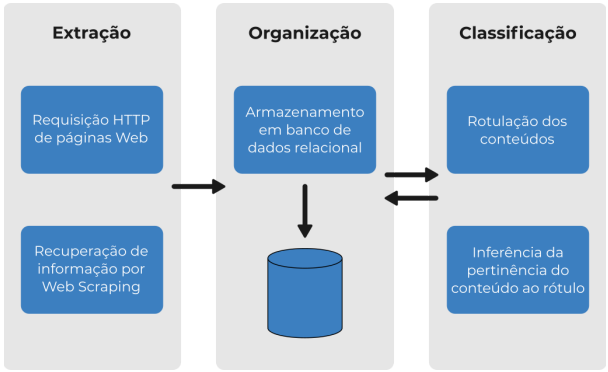

Figura 1: Diagrama de representação das responsabilidades dos módulos.

2.1 Módulo de Extração de Dados

O módulo de extração de dados é responsável por recuperar dados de páginas em hipertexto da internet. Para cumprir este objetivo, foi desenvolvido um algoritmo de Web Scraping (em português, Raspagem de Dados), que é uma técnica para extração de dados presentes em páginas web e conversão para um modelo estruturado de dados.

O processo de Raspagem de Dados pode ser dividido em três etapas: etapa de requisição, etapa de extração e etapa de transformação [6]. Na primeira etapa, o algoritmo requisita a página encontrada no Localizador Uniforme de Recursos (URL) utilizando o método GET. A requisição que utiliza o método GET solicita a representação de um recursos específico, retornando apenas dados [7].

A etapa de extração é quando, em posse dos dados não estruturados, são extraídas as informações requeridas por meio de filtros. Estes filtros podem ser implementados utilizando tags HTML, seletores CSS, expressões regulares ou mesmo a XML Path Language (XPATH), linguagem de consulta para navegar documentos XML e HTML.

Por fim, na etapa de transformação, os dados que foram extraídos podem ser transformados para apresentação ou armazenamento. Finalizada esta etapa, o módulo de extração deve enviar as informações coletadas para o módulo de organização.

O módulo em questão busca links em outros buscadores, utilizando conteúdos (descritos no Projeto Pedagógico de Curso de Automação Industrial) cadastrados em um banco de dados como os termos de busca. Para auxiliar nesta busca, também são utilizadas as tags pertencentes às disciplinas dos conteúdos buscados. Assim, se um conteúdo intitulado “Multiplexadores” é relacionado à disciplina “Eletrônica Digital”, e esta disciplina possui como tags “portas lógicas” e “circuitos”, os termos de busca utilizados são: “Multiplexadores”, “Multiplexadores portas lógicas” e “Multiplexadores circuitos”.

2.1 Módulo de Organização

Finalizada a captura de informações, é preciso armazená-las adequadamente. Para tal finalidade, utilizou-se uma estrutura de banco de dados relacional, permitindo a criação de tabelas que organizam dados a partir de relacionamentos predefinidos.

O banco de dados, apresentado na Figura 3, foi modelado de acordo com as cinco principais entidades do projeto, sendo elas a disciplina (subject), o conteúdo (content), o link (link), o voto (vote) e a tag (tag).

A disciplina, refere-se às matérias da matriz do Curso Técnico em Automação Industrial e pode se relacionar com múltiplos conteúdos. O conteúdo, que pertence a uma disciplina, pode se relacionar com outros conteúdos por meio de relacionamentos com chaves estrangeiras. Assim, um conteúdo pode se relacionar com um conteúdo posterior (Figura 3 - ID_Next_Content) ou um conteúdo anterior (Figura 3 - ID_Previous_Content). Quando um conteúdo possuir um ID_Next_Content, este conteúdo posterior será compreendido com um conteúdo filho do primeiro. Este primeiro torna-se um conteúdo pai do posterior, portanto sendo

referenciado através do ID_Previous_Content para o conteúdo filho.

Além disso, um conteúdo pode ter múltiplos links (provenientes das buscas realizadas no Módulo de Extração de Dados). Estes links, por sua vez, possuem múltiplos votos booleanos (atributo is_reliable da tabela Vote). Por fim, um conteúdo pode ter diferentes tags. Estas tags, como já citado, auxiliam na busca por links de um conteúdo específico.

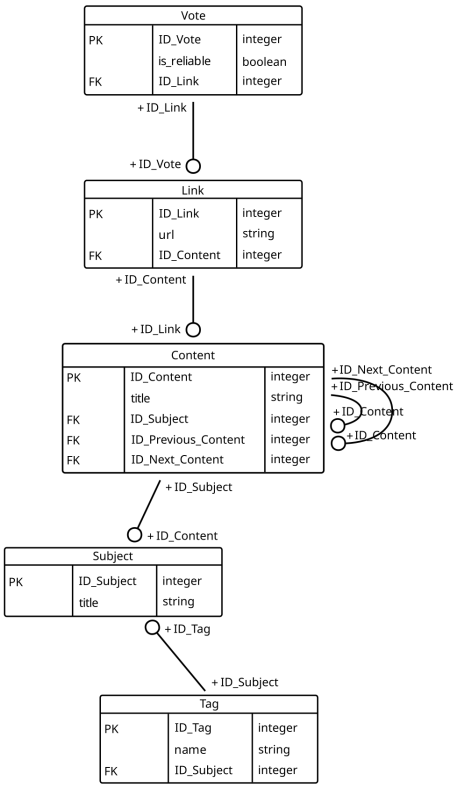

Figura 2: Diagrama de entidade-relacionamento do banco de dados.

Inicialmente, priorizou-se o desenvolvimento de uma estrutura mais enxuta com maior foco nas relações entre as tabelas, tornando possível uma rápida prototipagem e, consequentemente, viabilizando os testes necessários para adequação do módulo. Deste modo, os conteúdos das buscas são armazenados para permitir que o Módulo de Classificação acesse as informações de forma assíncrona.

2.1 Módulo de Classificação

Depois da organização e armazenamento das informações, é necessário um mecanismo de classificação automática dos conteúdos. O aprendizado de máquina (em inglês, machine learning) é um método de análise automática de dados que detecta padrões significativos em dados [8].

O aprendizado supervisionado ocorre por meio de uma função analítica que toma por base dados de treino em pares de entrada-saída. Ou seja, o algoritmo de aprendizado de máquina supervisionado utiliza um dataset de treino rotulado para realizar novas predições.

A estratégia de classificação do sistema combina o termo de busca inicial, a disciplina a qual este termo de busca pertence e as oito palavras que mais se repetem em um texto recuperado pelo módulo de extração. No caso do projeto em questão, o dataset de treino foi construído de acordo com esta estratégia. Assim, torna-se possível indicar os assuntos dos quais os conteúdos extraídos fazem parte.

O Quadro 1 ilustra, de forma didática, a estrutura do dataset de treinamento.

	Conteúdo 1	**Conteúdo 2**
Disciplina de busca	Metrologia	Controle
Termo inicial de busca	Mean Time Between Failures	Controle PID
1ª palavra mais repetida	tempo	controle
2ª palavra mais repetida	manutenção	pid
3ª palavra mais repetida	máquina	integral
4ª palavra mais repetida	paradas	velocidade
5ª palavra mais repetida	mttr	ação
6ª palavra mais repetida	mtbf	processo
7ª palavra mais repetida	indicadores	x
8ª palavra mais repetida	parada	proporcional
Classificação conteúdo	Mean Time Between Failures	Controle PID

Quadro 1: Modelo de dataset de treinamento com dois exemplos.

Para a realização de testes, a construção do dataset de treinamento levou em consideração dois conteúdos de disciplinas distintas. Inicialmente, foram eleitas manualmente as oito palavras mais importantes para cada conteúdo. Estas palavras foram permutadas

em todas as possibilidades de combinação entre as colunas de palavras mais repetidas, o que significa que cada conteúdo possui 40320 observações distintas. Assim, o dataset contém 80640 observações e uma linha de cabeçalho, totalizando 80641 linhas.

	A	B	C	D	E	F	G	H	I	J	
80621	Controle	Controle PID	processo	sistema	proporcional	ação	controle	integral	velocidade	pid	Controle PID
80622	Controle	Controle PID	processo	sistema	proporcional	ação	controle	velocidade	pid	integral	Controle PID
80623	Controle	Controle PID	processo	sistema	proporcional	ação	controle	velocidade	integral	pid	Controle PID
80624	Controle	Controle PID	processo	sistema	proporcional	ação	pid	controle	integral	velocidade	Controle PID
80625	Controle	Controle PID	processo	sistema	proporcional	ação	pid	controle	velocidade	integral	Controle PID
80626	Controle	Controle PID	processo	sistema	proporcional	ação	pid	integral	controle	velocidade	Controle PID
80627	Controle	Controle PID	processo	sistema	proporcional	ação	pid	integral	velocidade	controle	Controle PID
80628	Controle	Controle PID	processo	sistema	proporcional	ação	pid	velocidade	controle	integral	Controle PID
80629	Controle	Controle PID	processo	sistema	proporcional	ação	pid	velocidade	integral	controle	Controle PID
80630	Controle	Controle PID	processo	sistema	proporcional	ação	integral	controle	pid	velocidade	Controle PID
80631	Controle	Controle PID	processo	sistema	proporcional	ação	integral	controle	velocidade	pid	Controle PID
80632	Controle	Controle PID	processo	sistema	proporcional	ação	integral	pid	controle	velocidade	Controle PID
80633	Controle	Controle PID	processo	sistema	proporcional	ação	integral	pid	velocidade	controle	Controle PID
80634	Controle	Controle PID	processo	sistema	proporcional	ação	integral	velocidade	controle	pid	Controle PID
80635	Controle	Controle PID	processo	sistema	proporcional	ação	integral	velocidade	pid	controle	Controle PID
80636	Controle	Controle PID	processo	sistema	proporcional	ação	velocidade	controle	pid	integral	Controle PID
80637	Controle	Controle PID	processo	sistema	proporcional	ação	velocidade	controle	integral	pid	Controle PID
80638	Controle	Controle PID	processo	sistema	proporcional	ação	velocidade	pid	controle	integral	Controle PID
80639	Controle	Controle PID	processo	sistema	proporcional	ação	velocidade	pid	integral	controle	Controle PID
80640	Controle	Controle PID	processo	sistema	proporcional	ação	velocidade	integral	controle	pid	Controle PID
80641	Controle	Controle PID	processo	sistema	proporcional	ação	velocidade	integral	pid	controle	Controle PID

Figura 3: Visualização da seção final do dataset de treinamento.

A estratégia previamente citada utiliza de parte de um conjunto de técnicas de processamento de linguagem natural que permite aos computadores extrair sentido da linguagem natural humana.

Para contar as palavras que mais se repetem em um determinado texto, é preciso utilizar de técnicas de processamento de linguagem natural. Para o módulo em questão, foram implementadas funções auxiliares para a tokenização de textos, filtragem das palavras vazias e contagem dos tokens que mais se repetem. Estas funções foram implementadas em Python utilizando a biblioteca nltk.

Em primeira instância, foi implementada a função de tokenização do texto. Para alcançar tal objetivo, esta função recebe como parâmetro um texto a ser analisado. Este texto é percorrido linha por linha, cada linha é transformada inteiramente para caixa baixa e transformada em tokens com a ajuda da classe RegexpTokenizer da biblioteca nltk.

```
def get_word_list(text_file: TextIOWrapper) -> list:
    """Gets only the words inside a text file.

    Args:
        text_file (TextIOWrapper): content of a given file

    Returns:
        list: list of words inside a text file
    """
    word_list = list()
    reg_tokenizer = RegexpTokenizer(r"\w+")

    for line in text_file:
        lower_line = line.lower()
        word_list.extend(reg_tokenizer.tokenize(lower_line))

    return word_list
```

Algoritmo 1: Função que transforma input de texto em lista de palavras.

Uma vez em posse dos tokens, a etapa seguinte é a de filtrar as palavras vazias (em inglês, stop words). As palavras vazias são desconsideradas por não carregar valor semântico à análise.

```
def get_filtered_words(text_file: TextIOWrapper) -> list:
    """Filters words from stopwords inside a text file.

    Args:
        text_file (TextIOWrapper): content of a given file

    Returns:
        list: list of filtered words inside a text file
    """
    word_list = get_word_list(text_file)
    stop_words = set(stopwords.words("portuguese"))
    filtered_list = [word for word in word_list if word.casefold()
not in stop_words]
    return filtered_list
```

Algoritmo 2: Função que transforma input de texto em lista de palavras filtradas.

Por fim, podemos contar os tokens restantes que mais se repetem.

```
def get_common_words(text_file: TextIOWrapper, number:
int) -> list:
    """Get frequency of words inside a text file.

    Args:
        text_file (TextIOWrapper): content of a given file
        number (int): number of most commons words to be
retrieved

    Returns:
        list: list of the most frequent words inside a text file
    """
    filtered_list = get_filtered_words(text_file)
    word_frequency_list = FreqDist(filtered_list)

    return word_frequency_list.most_common(number)
```

Algoritmo 3: Função que transforma input de texto em lista de palavras mais frequentes com frequência em número.

Para o processo de classificação, implementou-se uma rotina em Python utilizando as bibliotecas scikit-learn e pandas. Em sua primeira etapa, a rotina importa os datasets de treino e predição que encontram-se em formato csv.

```
train_file = "train_dataset.csv"
predict_file = "predict_dataset.csv"

columns = ["DB", "TIB", "PF1", "PF2", "PF3", "PF4", "PF5",
```

```
"PF6", "PF7", "PF8"]

train_dataset = pd.read_csv(train_file, names=columns,
skiprows=1, delimiter=',')
predict_dataset = pd.read_csv(predict_file, names=columns,
skiprows=1, delimiter=',')
```

Algoritmo 4: Importação dos dados em formato csv

Após a devida importação dos dados, estes são tratados para que possam ser utilizados no modelo de classificação. Para atingir este objetivo, todos os valores são transformados para caixa baixa, os espaços em branco de palavras compostas são substituídos por um underscore e a palavra resultante é transformada em uma label codificada com um valor numérico. Esta última etapa é essencial para o funcionamento do modelo de aprendizado, uma vez que este só consegue lidar com valores numéricos. A transformação descrita pode ser visualizada no trecho de código a seguir.

```
label_encoder = LabelEncoder()

for column in columns:
    train_dataset[column] = label_encoder.fit_transform(
        train_dataset[column].apply(
            lambda x: x.strip().replace(' ', '_')).str.lower())
    predict_dataset[column] = label_encoder.fit_transform(
        predict_dataset[column].apply(
            lambda x: x.strip().replace(' ', '_')).str.lower())
```

Algoritmo 5: Transformação dos dados de cada coluna

Após a transformação dos dados, são separadas as colunas com os valores para a predição da coluna de predição em si, sendo esta a última coluna do dataset. Deste modo, é possível passar para o modelo de aprendizado de máquina os valores que levam a uma determinada classificação.

```
train_array = train_dataset.values
X_train = train_array[:, 0:9]
Y_train = train_array[:, 9]

predict_array = predict_dataset.values
X_predict = predict_array[:, 0:9]
Y_predict = predict_array[:, 9]
```

Algoritmo 6: Separação das colunas de treinamento da coluna de predição

Por fim, é construído um modelo de predição utilizando Árvore de Decisão. A escolha deste algoritmo de decisão se dá por sua boa capacidade em lidar com múltiplos dados categóricos.

```
model = DecisionTreeClassifier(criterion="entropy",
max_depth=3)

model.fit(X_train, Y_train)

predictions = model.predict(X_predict)
```

Algoritmo 7: Predição realizada com modelo de Árvore de Decisão.

Para que este modelo de classificação continue evoluindo, foi planejada a implementação de um website para que docentes e profissionais formados no Eixo Tecnológico de Controle e Processos Industriais possam avaliar as predições realizadas pelo Módulo de Classificação. Desta forma, ao avaliar como "confiável" ou "não confiável" uma predição do algoritmo, o website insere uma informação na tabela Voto à respeito do link. Este voto é somado aos demais para um momento de disponibilização posterior. Assim, será possível avaliar as classificações realizadas de modo a sinalizar as necessidades de ajuste no dataset de treinamento.

Figura 4: Representação da tela de votação proposta em arte gráfica.

3 CONCLUSÕES

Verifica-se que a utilização do IRIS automatiza processos repetitivos de busca por conteúdos técnicos e, portanto, contribui para a melhor formação do Técnico em Automação. Desta forma, o sistema posiciona-se como uma valiosa TIC no ambiente escolar, permitindo que os docentes utilizem desta ferramenta para tornar o ensino mais estimulante.

Diante disso, espera-se que o sistema proposto fortaleça a autonomia nos estudos dos discentes, colaborando para a sua permanência e êxito nas instituições de ensino brasileiras.

Planeja-se, no futuro, a implementação de um módulo para disponibilizar de maneira eficiente as informações pesquisadas de modo a reduzir ainda mais o tempo gasto em processos repetitivos de buscas manuais por informação.

REFERÊNCIAS

[1] Ministério da Educação. 2022. Catálogo Nacional de Cursos Técnicos: Técnico em Automação Industrial. Disponível em: http://cnct.mec.gov.br/cursos/curso?id=29

[2] Karine Rodrigues Alvarez; Suelem Cristina Alves; Roberta Pereira Matos. 2021.Evasão escolar nos cursos técnicos integrados ao ensino médio da Rede Federal: Levantamento de fatores motivacionais e propostas de intervenção. Disponível em: https://rsdjournal.org/index.php/rsd/article/view/15630/13933

[3] Comitê Gestor de Internet no Brasil. 2020. TIC Educação: Pesquisa Sobre o Uso das Tecnologias de Informação e Comunicação nas Escolas Brasileiras. São Paulo. Disponível em: https://www.cetic.br/media/docs/publicacoes/2/20201123090444/tic_edu_2019_livro_eletronico.pdf

[4] Oren Etzioni. 2011. Search needs a shake-up. Revista Nature. Disponível em: https://www.nature.com/articles/476025a

[5] Nações Unidas Brasil. 2022. Objetivos de Desenvolvimento Sustentável. Disponível em: https://brasil.un.org/pt-br/sdgs

[6] Emil Persson. 2019. Evaluating tools and techniques for web scraping. Disponível em: https://www.diva-portal.org/smash/get/diva2:1415998/FULLTEXT01.p

[7] Mozilla Foundation. 2022. Métodos de requisição HTTP. Disponível em: https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Methods

[8] Shai Shalev-Shwartz e Shai Ben-David. 2014. Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press. Disponível em: https://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning/understanding-machine-learning-theory-algorithms.pdf